SUPER sinc

n° 4 Ottobre 1984

L. 7.500

Spedizione in abbonamento postale Gruppo III/70

4

Una pubblicazione della J.soft EDITRICE

in collaborazione con GRUPPO EDITORIALE JACKSON

SUPERSINC
la rivista per gli utenti dei prodotti SINCLAIR



COSA CAMBIA CON I MICRODRIVE?

FORMULA 1: UN GRANDE GIOCO PER LO ZX81

COME USARE I CARATTERI GRAFICI

WARLOCK

ZX81 GRAFICA

GALATTOIDI IN L/M

I LIBRI: FURTHER PROGRAMMING FOR THE ZX SPECTRUM



Allegre,
Fresche,
Spiritose,
Pratiche.
HOME COMPUTER
hc
LA RIVISTA DEL COMPUTER IN CASA
UNA PUBBLICAZIONE DEL GRUPPO EDITORIALE JACKSON
OTTOBRE 1984 L. 3.500
13
3 PROGRAMMI PER
VI SVELIAMO I
COSTRUIAMO INSIEME
METTI L'ATLANTE N
VIC20, SEGA, SPECTRUM, TI99/4A, CBM64, A
VIDEO GIOCHI 19
ISSN: 0392-8918
UNA PUBBLICAZIONE DEL GRUPPO EDITORIALE JACKSON
LA PRIMA E UNICA RIVISTA ITALIANA DI VIDEOGAMES E GIOCOMPUTER
OTTOBRE 1984 L. 3.500
Spedizione in Abb. Postale Gruppo III/70
I FUMETTI
COL COMPUTER
14 NOVITÀ IN PROVA
NUOVO
COMPUTER ELECTRON
TUTTI I RECORD
DEI CAMPIONI
Con tutta la competenza del
GRUPPO EDITORIALE
JACKSON
JACKSON

SUPER SINC

J. soft s.r.l
DIREZIONE, REDAZIONE, AMMINISTRAZIONE
Via Rosellini, 12
20124 MILANO
Tel. (02) 68.88.228

DIRETTORE RESPONSABILE:
Pietro Dell'Orco
COORDINAMENTO TECNICO:
Riccardo Paolillo
REDAZIONE:
Lucio Bragagnolo
Marcello Spero
HANNO COLLABORATO A QUESTO NUMERO
Alberto Chiericati
Carlo Corti
Mauro Corti
Enrico Levantino
Paolo Maffei
Massimo Magnani
Angelo Motta
Carlo Panzalis
Francesco Santi

GRAFICA E IMPAGINAZIONE:
Margherita La Noce
Raffaella Toffolatti

FOTOCOMPOSIZIONE:
d&b Via Vignola, 5
Tel. 02/59.85.08
20133 MILANO

CONTABILITÀ:
Giulia Pedrazzini
Flavia Bonaiti

AUTORIZZAZIONE ALLA PUBBLICAZIONE:
Tribunale di Milano n° 201
del 14.04.1984

STAMPA:
Litografia del Sole
Albairate (MI)

PUBBLICITÀ
Concessionario per l'Italia e l'Estero Reina s.r.l.
Via Washington, 50
20046 MILANO
Tel. (02) 49.88.066 (5 linee R.A.)
Tlx. 316213 REINA 1

Concessionario esclusivo per la DIFFUSIONE in Italia e Estero:
SODIP - Via Zuretti, 25
20125 MILANO

Spedizione in abbonamento postale Gruppo III/70
Prezzo della rivista L. 7.500
Numero arretrato L. 15.000
Abbonamento annuo (11 numeri) L. 75.000; per l'estero L. 110.000
i versamenti devono essere indirizzati a:
J.soft s.r.l.
Via Rosellini, 12
20124 MILANO
mediante emissione di assegno bancario, cartolina, vaglia o utilizzando il c/c postale
n. 19445204
Per i cambi di indirizzo indicare, oltre al nuovo, anche l'indirizzo precedente ed allegare alla comunicazione l'importo di L. 500, anche in francobolli

GLI ARTICOLI TRADOTTI SONO TRATTI DALLA RIVISTA
ZX COMPUTING - ARGUS SPECIALIST PUBLICATION L.T.D.
1 GOLDEN SQUARE LONDON W1R3AB
SINCLAIR USER/SINCLAIR PROGRAMS
ECC PUBLICATIONS
196-200 BALLS POND ROAD
LONDON N14AQ

Nel prossimo numero

REM:HW — L'interfaccia come e perché: lo Spectrum diventa «computer di casa».

LOAD — Formula uno: il disassemblato del programma

ZX81: disegno in prospettiva

UDG: la seconda puntata, più...

Generatore di caratteri per Spectrum

Caratteri grafici e linguaggio macchina

PRINT — I libri da leggere...

R·U·N — ...e il software da giocare

ZX CLUB — Notizie e programmi dall'Italia tutta

POINT

Pirateggiando

Sempre con maggiore frequenza si assiste al "fenomeno", diciamo così, della pirateria del software. Non che la cosa sia nuova, vale comunque la pena di spendere qualche parola a tal proposito.

Per scrivere del buon software giocoso, professionale o gestionale servono molte ore di duro lavoro intellettivo, un certo numero di ore di lavoro manuale relative alla codifica, oltre al tempo necessario per il test e la correzione del programma.

Forse qualcuno di voi si è cimentato nella scrittura di programmi semplici o complessi e si sarà reso conto di quanta fatica sia necessaria per la messa a punto di un prodotto software.

Data la premessa è chiaro che pirateggiare, rubando il software altrui per venderlo sottocosto è azione degna perlomeno di biasimo se non di azioni legali vere e proprie.

La legislazione italiana non prevede ancora forme di protezione efficaci per chi cerca di vivere ed operare onestamente nel settore del software; di conseguenza individui privi di scrupoli e di etica professionale possono "rubare" impunemente il prodotto altrui frustrando così ogni possibilità di onesti guadagni.

Un vasto numero di operatori del settore sta organizzando iniziative per reprimere questo dilagante malcostume. Per ogni forma di protezione del software messa a punto dagli autori vengono prodotti altrettanti sistemi di copiatura abusiva.

Una situazione senza fine perpetrata in continuazione a scapito di una qualificata offerta di validi prodotti.

Tutto ciò crea la necessità di elevare i prezzi per recuperare, almeno in parte, l'investimento "tempo" speso per l'elaborazione del programma. I maggiori costi gravano, assurdamente, sugli utenti onesti che acquistano regolarmente il programma.

Una cassetta con un buon gioco per VIC, C64 o Spectrum potrebbe costare dalle 3 alle 5.000 lire se questo mercato seguisse le normali regole del commercio.

La prossima volta che qualcuno vi propone l'acquisto di software copiato, meditate un attimo al danno provocato da tale gesto e forse, in futuro, allo stesso prezzo potrete comprare una copia originale del programma.

Pietro Dell'Orco

NON UN ABBONAMENTO!

UN SUPERABBONAMENTO

Questo è ciò che la J.*soft* ti offre: un superabbonamento ad una super-rivista! Una eccezzionale offerta di lancio: 13 numeri di SUPERSINC al prezzo di 10 oltre alla possibilità esclusiva di ricevere mensilmente, con la copia della rivista, il supporto magnetico con tutti i programmi pubblicati. L'offerta è valida fino al 20/10/1984.



LE NOSTRE SUPERPROPOSTE ABBONAMENTI

Abbonamento* a 13 numeri (2 nel 1984 + 11 nel 1985) **di SUPERSINC** al prezzo speciale di L. 35.000

Abbonamento* a 13 numeri di SUPERSINC + 13 cassette con tutti i programmi pubblicati al prezzo speciale di L. 75.000

***L'abbonamento decorrerà dal prossimo numero**

Abbonarsi è semplice! Effettuate il versamento con l'apposito modulo c.c.p. inserito in questo fascicolo, oppure ritagliate il tagliando abbonamenti pubblicato in questa pagina e speditelo allegando un assegno intestato a:
J.*soft* - Via Rosellini 12 - 20124 Milano.

Tagliando abbonamento a SUPERSINC da inviare in busta chiusa a:
J.*soft* - Via Rosellini 12 - 20124 Milano

☐ Desidero sottoscrivere un abbonamento a 13 numeri (2 nel 1984 + 11 nel 1985) di SUPERSINC al prezzo speciale di L. 35.000

☐ Desidero sottoscrivere un abbonamento a 13 numeri (2 nel 1984 + 11 nel 1985) di SUPERSINC + 13 cassette con tutti i programmi pubblicati al prezzo speciale di L. 75.000

cognome ____________________ nome ____________________

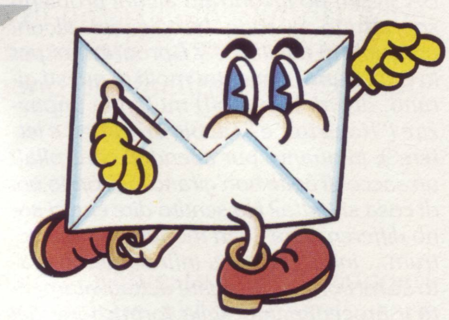
via ____________________ città ____________________

cap. __________ provincia __________ data __________

firma ____________________

READ &WRITE

La posta dei lettori



Stampa o stampante?

Nei listati dovreste usare una vera stampante (magari un'Alphacom)... mantenete un filo diretto con l'Inghilterra, visto che lì le novità hard e soft (soprattutto soft) sono costanti.

Andrea Rampi - Borgolavezzaro (NO)

Questa lettera è solo un esempio delle tante che ci sono pervenute in merito alla questione della "stampa" dei listati, su cui la maggioranza dei lettori mostra delle perplessità (soprattutto per la probabilità più alta di commettere errori di composizone). A questo proposito diciamo a tutti che i listati da noi pubblicati NON sono stampati, bensì, appunto, "listati" nel senso letterale del termine da una stampante professionale interfacciata con uno dei nostri Spectrum: ragion per cui non avete da temere nessun errore in quanto i programmi, una volta verificati, vengono listati direttamente senza passaggi intermedi. Dopo di che l'output su carta viene fotografato (NON composto) e riprodotto sulla rivista. Su SUPERSINC leggete i listati dei programmi esattamente come escono dalla nostra stampante, senza manipolazioni di qualsiasi genere. Quanto al "filo diretto", la J. soft ha di recente concluso accordi in base ai quali ci troviamo da alcuni giorni in possesso di 200 programmi selezionati tra i migliori sul mercato britannico, e che vi presenteremo numero per numero. Oltre a questo, ci verrano fornite, ogni mese, le ultimissime novità, così che saremo in grado di darvi anche qualche anterpima (come è già successo, ad esempio, per "Cosmic Cruiser"). Vedrete che le sorprese non mancheranno!

Problemi di carattere... grafico

Caro SUPERSINC,
possiedo uno Spectrum 48K e ho un problema che mi assilla da mesi. Per definire un carattere grafico uso il sistema binario e guardando i vostri listati mi sono accorto che voi usate il seguente modo (l'esempio che riporto è tratto dal gioco "attacco missilistico"):

```
10 FOR n=0 TO 39
20 READ d: POKE USR "a"+n,d
25 NEXT n
1000 DATA 0,6,6,2,14,150,230,254,
230... e così via per un totale di 40 numeri.
```

Ora vorrei sapere: 1) Come fate a calcolare i DATA (per esempio, 238)? 2) Il 39 della linea 10 da cosa dipende?

Francesco Pertoldi - Roma

238 (e tutte le altre cifre) sono la traduzione in decimale (cioè quella che usiamo correntemente) dei loro corrispettivi in binario; dal punto di vista della definzione dei caratteri grafici, il calcolatore accetta indifferentemente una forma o l'altra — e difatti, nel caso specifico, 238 equivale a scrivere BIN 11101110. La conversione dal binario al decimale si ottiene moltiplicando ogni cifra per una potenza crescente di 2 (partendo da destra, il primo numero va moltiplicato per 2 elevato allo zero, il secondo per 2 elevato alla prima, eccetera) e sommando tutti i valori così calcolati. Il ciclo della linea 10 va da 0 a 39, dando quindi progressivamente a "n" 40 valori. Ognuno di questi corrisponde a uno dei 40 numeri delle linee DATA di cui sopra: come abbiamo già detto, ogni cifra equivale a un numero binario che, nel sistema di definizione dei caratteri dello Spectrum, definisce un ottavo (una riga) di carattere. Utilizzando quindi otto numeri per ognuno di essi, con 40 numeri definieremo 5 caratteri grafici. Per ulteriori spiegazioni rimandiamo tutti alla lettura di "UDG: ovvero...." di Carlo Panzalis (di cui pubblichiamo in questo numero la prima parte), articolo ricco di chiarimenti e approfondimenti sul tema.

Per lo ZX81

Vi raccomando di mantenere la promessa di inserire nella rivista programmi brevi anche per lo ZX81.

Giovanni Patané - Genova

Come puoi vedere la promessa è stata mantenuta, anche se non riguardava la... brevità dei programmi. In ogni caso, per gli amanti della concisione, contiamo di pubblicare nei prossimi numeri anche alcuni listati per lo ZX81 inespanso (che è dotato, come è noto, di 1K RAM).

L'avventura degli "Adventure"

Sono un ragazzo di quindici anni appassionato di computer e videogames. Posseggo lo ZX Spectrum e devo dire che sono molto soddisfatto delle sue prestazioni. Vorrei farvi un paio di domande: 1) Potete darmi delle spiegazioni sui giochi "Adventure" (ne ho uno ma non so farlo funzionare: aiutatemi!!!)?
2) Ho saputo che sono usciti giochi dell'Atarisoft che vanno bene anche sul Sinclair. È vero ?

Lettera firmata

Carissimo,
innanzitutto una raccomandazione: la prossima volta metti il tuo nome non sulla busta (che viene trattata in genere piuttosto male dalle nostre segretarie) ma anche sulla lettera, OK? Venendo alle tue domande, 1) Per poterti dare dei consigli avremmo bisogno di ricevere qualche informazione in più; ci interesserebbe soprattutto sapere qual è l'avventura che ti crea questi problemi. Per consigli generici, invece, non è questa la sede adatta: vedremo invece di pubblicare uno o più articoli sull'argomento. 2) Verissimo. Potrai trovarli in qualunque negozio specializzato e aggiornato.

READ &WRITE

Erroracci ed errorini

Cari amici di SUPERSINC,
sono uno studentello alle prime armi col BASIC ed ho incontrato alcuni problemi soprattutto quando ho digitato alcuni programmi dal libro "77 programmi per lo Spectrum", e mentre molti di questi girano, due di loro (ahi!) mi fanno impazzire ("Roulette" e "Tavole di numeri e lettere"), in quanto pur avendoli controllati un sacco di volte non girano. Secondo voi di cosa si tratta? Ho sentito dire che ci sono differenze tra i vari modelli dello Spectrum... inoltre vi sarei infinitamente grato se mi vorreste spiegare dettagliatamente il procedimento sulla formazione dei caratteri definiti dall'utente, in quanto seguendo le istruzioni fornite dal manuale in versione italiana a pagina 145 non succede niente.

Giorgio Mangoni - Napoli

Quei due programmi sono sicuramente esatti (del resto, ne vengono anche pubblicate rispettivamente una schermata e l'hard copy su stampante), e gli eventuali malfunzionamenti sono da attribuire ad errori di copiatura. Per "Roulette" sappiamo che molti commettono l'errore di scrivere la parte di stringa "()" — all'interno della linea 130 — usando il tasto "w" con SYMBOL SHIFT invece di utilizzare, sempre con SYMBOL SHIFT, i tasti "r" e "t". Quanto al secondo, le indicazioni forniteci sono insufficienti per azzardare ipotesi, né ci sono pervenute altre segnalazioni da parte dei lettori. Consigliamo comunque di non cambiare in minuscole (specie in "Roulette") le lettere maiuscole presenti nelle varie stringhe alfanumeriche.
In ogni caso, non si tratta di inconvenienti derivanti dall'uso di Spectrum di versioni diverse da quella con cui il programma è stato scritto. Le uniche differenze tra le versioni due e tre dello Spectrum (quelle in vendita in Italia) riguardano l'uso della funzione di lettura della tastiera IN, che ritorna valori diversi a seconda della versione. Da qualche tempo si trovano in commercio anche due ulteriori elaborazioni della versione tre, chiamate 3a e 3b, sulle quali non abbiamo per ora informazioni (tenete sempre presente l'anticipo con cui viene redatta la rubrica).
Infine, l'unico errore presente nella spiegazione del manuale è quello di dare un carattere grafico definito come formato da... nove linee BIN anziché otto. Il resto del testo è corretto: se tutte le operazioni descritte vengono compiute senza commettere errori — ricordarsi di dare il RUN al programmino di inserimento! —, entrando in modo grafico e digitando la "p" deve apparire il simbolo grafico definito (la linea BIN da tralasciare è una qualsiasi di quelle tra la quinta e l'ottava compresa), cioè pi greco.

Quel QL...

Suggerirei di aprire su SUPERSINC una rubrica dedicata al QL Sinclair, con qualche piccolo listato...

Stefano Pini - Vedano al Lambro (MI)

Certamente non mancheremo di dedicare uno, anzi, più di uno spazio all'ultimo calcolatore Sinclair. Nel frattempo, invitiamo i (per il momento) pochi lettori che lo posseggono a fornirci notizie, curiosità, bug e... anche listati.

Perché niente sport?

Caro SUPERSINC,
perché sullo Spectrum non esistono il calcio e altri sport tipo Commodore? Spero molto in una risposta soddisfacente.

Osfaldo Pagano - Catania

Lo Spectrum si presta un po' meno di altri computer ("tipo Commodore") alla simulazione sportiva, soprattutto per limiti intrinseci riguardanti alcune particolarità grafiche e, più in generale, la velocità. Tuttavia, ciò non è sufficiente a giustificare una lacuna così grande, tanto più che il software disponibile per lo ZX non è certo poco... che dire? Innanzitutto giriamo la proposta alle software house e ai programmatori più capaci; in secondo luogo, segnaliamo che da qualche tempo sono scese in campo, per quanto riguarda i programmi per Spectrum, la Atarisoft e la Activision, società notissime nel campo dei videogiochi da consolle. La seconda ha già creato un Tennis per Spectrum: e conoscendo l'inclinazione di queste due "grandi" verso i giochi ad argomento sportivo, possiamo ragionevolmente sperare che i loro progetti si sviluppino anche in questa direzione. Soddisfatto?

Benedetti manuali!

Ha da poco comprato lo ZX, e leggendo il manuale non ho capito a cosa servono le POKE, PEEK e DATA, che mi pare siano molto importanti.

Franco Spedale - Desenzano (BS)

Ma davvero il manuale si presenta così oscuro? Comunque, l'istruzione POKE serve ad introdurre o modificare valori numerici presenti nelle singole locazioni di memoria: per esempio, POKE 23609,255 inserisce il valore 255 all'interno della locazione 23609, che per inciso regola il... beh, chi non lo sa già provi a vedere! PEEK serve a leggere il valore di una locazione: nel caso di cui sopra, è possibile sapere che valore è stato inserito nella locazione 23609 facendo, appunto, PRINT PEEK 23609, che visualizzerà il valore 255 (o il numero in quel momento presente in 23609). Quanto a DATA, quest'istruzione serve a inserire in un programma liste di numeri, o stringhe alfanumeriche, che verranno lette dal programma stesso al momento opportuno mediante l'istruzione READ. Non ci addentriamo in spiegazioni più approfondite, non essendo questa la sede più opportuna; vi diciamo però che stiamo realizzando una rubrica che analizzerà approfonditamente un comando alla volta di quelli disponibili nel Basic Sinclair, che speriamo potrà chiarire dubbi e interrogativi a chi non riesce a districarsi nelle (non sempre, però) un po' ermetiche spiegazioni del manuale (specie quello in edizione italiana, dove figura qualche piccolo errore assente nell'edizione originale).

La nave del destino

Spettabile Redazione,
sarei lieto se potesse pubblicare questo appello che riguarda la cassetta-gioco "The Ship of Doom", un adventure game che ho avuto in omaggio insieme all'espansione di memoria per lo Spectrum: come si fa a prendere la chiave da sotto la campana di vetro? Ho provato col laser (che ho trovato in una stanza), ma dopo aver dato il comando "spara sul vetro" il computer risponde che "tutto l'involucro è stato distrutto" e non c'è più traccia della chiave, indispensabile per proseguire. Che senso ha la frase "non ti fidare delle donne" che il calcolatore dà come risposta al comando "help" quando si è nella stanza col corpo congelato?

Giovanni Amietta - Milano

Purtroppo non abbiamo mai avuto occasione di giocare a "The Ship of Doom", e quindi non possiamo essere molto di aiuto, tranne per girare l'appello a chi ci legge. Abbiamo compiuto un rapido sondaggio tra redazione e ufficio grafico per saperne di più circa il senso della frase "non ti fidare delle donne", ma i risultati ottenuti sono completamente impubblicabili....

Per collaborare a SUPERSINC

La maggior parte dei seguenti suggerimenti ha lo scopo di migliorare l'accuratezza e la velocità di pubblicazione di un articolo; rispettando questi consigli si accresceranno le probabilità che un vostro lavoro venga pubblicato. La rivista è interessata ad articoli e programmi riguardanti la linea Sinclair. Siamo più interessati al contenuto di un articolo piuttosto che al suo stile, e soprattutto gli articoli devono essere chiari ed esaurienti. La seguente guida permetterà che le vostre buone idee e i vostri programmi vengano più facilmente accettati per la pubblicazione:

1 l'angolo superiore sinistro della prima pagina dovrà contenere: nome, cognome, indirizzo, numero telefonico, codice fiscale e data di spedizione, luogo e data di nascita.

2 l'angolo superiore destro della prima pagina dovrà contenere il nome del computer al quale il lavoro si riferisce, unitamente alla configurazione richiesta (memoria occorrente, eventuali periferiche e così via).

3 il titolo sottolineato dell'articolo dovrà iniziare a circa due terzi in altezza della prima pagina.

4 le pagine seguenti potranno essere battute normalmente, con la condizione che l'angolo superiore destro contenga un'abbreviazione del titolo e del cognome, unitamente al numero di pagina progressivo. Per esempio, Horace Goes.../Brambilla/2.

5 tutte le linee del testo dell'articolo dovranno essere battute con spazio 2 o spazio 3, e un margine di circa un centimetro dovrà trovarsi ad entrambi i lati dello scritto.

6 dovrà essere usata una carta formato A4 e lo scritto dovrà occupare un solo lato del foglio (caratteri maiuscoli e minuscoli).

7 i fogli dovranno essere uniti con una clip.

8 avendo intenzione di spedire più di un articolo, questi dovranno essere inviati separatamente insieme alla rispettiva copia su supporto magnetico.

9 programmi brevi (meno di 20 linee) potranno essere inseriti nel testo, mentre programmi più lunghi dovranno essere listati separatamente. È **ESSENZIALE** per noi disporre di una copia del programma registrata più volte su supporto magnetico, su entrambi i lati dello stesso. È preferibile usare nastri di buona qualità e di lunghezza non eccessiva; la cassetta o la cartuccia per Microdrive dovranno essere etichettati con il nome dell'autore, il titolo dell'articolo, il computer interessato e soprattutto le eventuali espansioni richieste. Come suggerimenti di programmazione, si consiglia di usare, per esigenze di stampa listati, le istruzioni INK, PAPER, INVERSE piuttosto che scrivere direttamente in INVERSE VIDEO. Un rapido controllo dei programmi per operare queste sostituzioni sarà da noi estremamente apprezzato.

10 per maggior chiarezza, all'interno dell'articolo è conveniente usare caratteri maiuscoli riferendosi a istruzioni BASIC (esempio RETURN, LIST, RND, PRINT etc.). Se si desidera evidenziare una parola, è preferibile sottolinearla piuttosto che scriverla in carattere maiuscolo.

11 gli articoli ed i programmi potranno avere qualsiasi lunghezza — da una routine di una sola linea fino a programmi molto complessi.

12 volendo includere fotografie, queste dovranno essere in formato 24 x 36, o 6 x 6, in bianco e nero o diapositive.

13 non prenderemo in considerazione articoli che siano stati sottoposti ad altre case editrici.

14 il compenso per la collaborazione prestata sarà commisurato alla complessità e all'interesse del programma (da un minimo di L. **50.000** a un massimo di L. **300.000**). Il pagamento è effettuato in caso di pubblicazione del lavoro.

15 il materiale ricevuto e non pubblicato non verrà restituito.

Spedite i vostri lavori a:

SUPERSINC
Via Rosellini, 12
20124 Milano

e saremo lietissimi di pubblicare i contributi migliori.

La Redazione

OPEN

Cosa cambia con i Microdrive?

di **Marcello Spero**

Illustrazione e critica della rivoluzionaria (e discussa) memoria di massa sviluppata dalla Sinclair



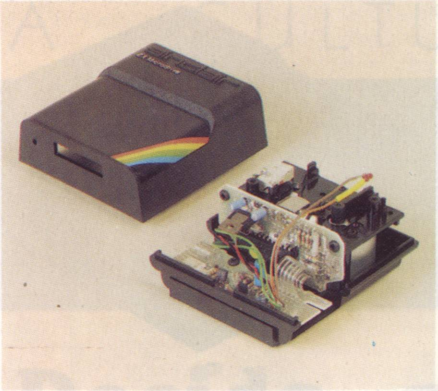
Ormai facilmente reperibili in tutta Italia, i famosi e tanto attesi Microdrive hanno trovato un folto pubblico di entusiasti ed uno, forse ancora più numeroso, di dubbiosi.
Molti sono infatti coloro che si chiedono se non si tratti, dopotutto, di una specie di "super registratore"; di qualcosa che, comunque, non giustifica una spesa di oltre trecentomila lire (per una sola unità).
Non è certamente nostra intenzione decidere per voi se sia il caso o meno di dotare il vostro Spectrum di questa nuova periferica; cercheremo soltanto di fornirvi gli elementi, che riteniamo obiettivi poiché provenienti da un'esperienza pratica di ormai quattro mesi, perché possiate giudicare voi stessi.

Cosa sono

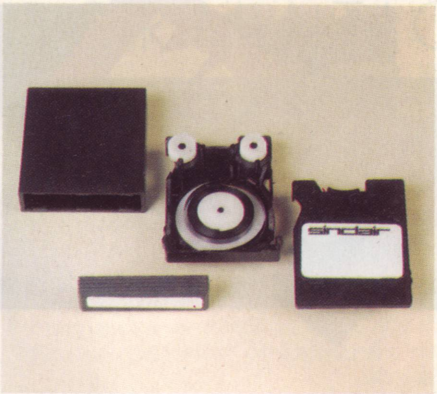
Nel corso della lunga gestazione dei Microdrive una ridda di informazioni, provenienti da fonti attendibili o presunte tali, ha contribuito a confondere non poco le idee di noi poveri mortali in attesa. Quando, finalmente, abbiamo avuto in mano il prodotto funzionante, ci siamo resi conto che dalla prima idea di Clive Sinclair molta acqua era passata sotto i ponti del Tamigi, e molte cose erano cambiate.
Il sistema Microdrive-interfaccia 1 (il primo non può infatti funzionare senza la seconda) è una periferica per l'archiviazione di masse di dati, cioè una memoria di massa, con velocità di funzionamento (cioè nel "registrare" e "restituire" le informazioni) e flessibilità di uso (cioè il numero di modi diversi in cui le informazioni possono essere organizzate nel suo interno) di gran lunga più elevate rispetto ad un registratore a cassette.

Come sono fatti

Il Microdrive in sé per sé, cioè il "pacchetto di sigarette", rappresenta la parte meccanica del sistema; è sostanzialmente un registratore molto veloce, che usa cassette senza fine (con gli estremi del nastro uniti) come il vecchio "stereo 8" di buona memoria.
Le cassettine, chiamate cartucce, hanno una struttura interna veramente semplice anche se, a nostro giudizio, non completamente affidabile. Il nastro, lungo circa 5 metri ed alto solo 1,5 mm., è unito ad anello ed avvolto su di una bobina in nylon che presenta inferiormente un largo piatto di contenimento, mentre superiormente ha un bordo inclinato. Esso viene dunque preso dal centro della bobina, posizione in cui sporge dall'alto grazie al bordo inclinato; passa attorno ad una prima rotella e quindi, tenuto teso da un feltrino a molla come quelli delle cassette, è portato anteriormente a contatto con la testina del Microdrive. A questo punto incontra una seconda rotella, che sporge dall'involucro della cartuccia, e sulla quale fa presa il capstan in gomma del motore. Infine rientra e viene riavvolto sulla parte esterna della bobina.
Come abbiamo detto, questo sistema non sembra dare eccessive garanzie; prova ne è il blocco di una delle nostre cartucce, causato da un tremendo e irrimediabile "arrotolamento" interno del nastro. Bisogna comunque dire che si è trattato di un caso isolato, e potremmo anche essere stati particolarmente sfortunati.
Oltre a non dover essere mai riavvolto, il nastro scorre con velocità tale da farlo passare tutto davanti alla testina ogni sette secondi circa: è questo perciò il tempo massimo di ricerca.
La parte elettronica che pilota il Micro-







drive è contenuta nell'interfaccia 1, che come senz'altro saprete è di forma piatta e si colloca al di sotto dello Spectrum, inclinandolo in avanti. Collegata al computer tramite il solito connettore a pettine posteriore, si collega al primo Microdrive per mezzo di un cavo piatto flessibile a 16 poli (un po' corto, in verità); le eventuali ulteriori unità, fino ad un totale di otto, vengono collegate l'una all'altra in modo rigido, per mezzo di una doppia spina, e bloccate con una staffa avvitata, per evitare improvvisi distacchi. Le possibilità dell'interfaccia 1 non si limitano comunque alla gestione dei Microdrive; in essa sono compresi anche due mezzi di collegamento con l'esterno: un'interfaccia seriale RS232 ed una per il collegamento di più Spectrum in rete locale. Di queste possibilità parleremo comunque in un prossimo articolo, riservando questo al Microdrive.

Come funzionano

Il collegamento del complesso Microdrive-interfaccia 1 rende operativi tutti quei comandi, posti sotto la prima fila di tasti, che finora davano, se usati, errori di vario tipo; inoltre consente l'uso di alcuni dei normali comandi in forme nuove, prima non accettate. Dei nuovi comandi parleremo fra un attimo; ora ci interessa analizzare come sia stato possibile rendere operativo qualcosa che per il BASIC dello Spectrum era errore.
La sintassi dei nuovi comandi, nonché delle nuove versioni di quelli "vecchi", è contenuta, insieme ad una valanga di informazioni riguardanti la manipolazione dei dati da e per i Microdrive e le altre linee (RS232 e rete), in una ROM da 8K contenuta nell'interfaccia 1.
Il problema di far "convivere" queste due ROM è stato risolto in un modo a dir poco originale; ogni volta che la CPU dello Spectrum tenta di eseguire le routine della ROM (quella originale) all'indirizzo 0008 esadecimale (si tratta della routine di errore) o 1708 esadecimale (riguarda la chiusura di un canale), l'interfaccia 1 se ne accorge e porta a livello alto la linea ROMCS, facendogli così letteralmente "sparire" la ROM originale della memoria. Al suo posto collega la nuova ROM, detta appunto "fantasma"; se l'errore che stava per essere segnalato con un messaggio della vecchia ROM riguarda uno dei nuovi comandi, o la chiusura di un canale riguardante il complesso interfaccia-Microdrive, la ROM fantasma resta collegata per il tempo necessario a compiere tutte le operazioni relative. Terminato il suo compito, le viene nuovamente sostituita la ROM originale, che si occuperà dell'istruzione successiva. Nel caso l'errore sia invece "davvero" un errore, che oltretutto non riguarda i nuovi comandi, il ritorno alla vecchia ROM è immediato, e la routine di errore riprende.
Questo sistema mette a disposizione dello Spectrum una quantità notevole (8K) di nuovo "materiale" su cui lavorare. Per gli errori riguardanti le nuove operazioni possibili, infatti, è presente una valanga di messaggi, che coprono ogni eventualità. La particolare struttura di analisi degli errori della nuova ROM, inoltre, consente al programmatore esperto (ma tutti possono diventarlo...) l'aggiunta di nuove istruzioni da lui create; di questo ci occuperemo diffusamente in futuro, dando la possibilità di usufruire di qualche utile istruzione extra anche a quanti non hanno la voglia o il coraggio di cimentarsi nell'impresa.
Malgrado una struttura operativa tanto complessa, il funzionamento dello Spectrum esteso (cioé con interfaccia e Microdrive) è estremamente trasparente per l'utilizzatore. Non ci è mai capitato che un file (cioé un blocco) di dati o di programma fosse stato registrato in modo difettoso. Può succedere invece, ma si tratta di un problema senza grande importanza, che in una cartuccia quasi piena un file non venga trovato che dopo un certo numero di passaggi; ci è anche capitato di ottenere dei cataloghi (maggiori spiegazioni su questo fra poco), con cartucce piene di un gran numero di file molto corti, in cui mancavano i nomi di alcuni file, peraltro regolarmente registrati e recuperabili.
Per quanto riguarda la durata delle cartucce, un argomento che ci aveva trovato dubbiosi, non c'é proprio niente da eccepire: il nostro accanimento si è infatti concentrato su una cassetta, che è stata usata per una media di cinque operazioni di i/o al giorno, per cinque giorni alla settimana, per quattro mesi (escluse le ferie, naturalmente!), senza dare ancora nessun segno di "stanchezza", che il manuale descrive come un generale allungamento nei tempi di ricerca.

Come si usano

La prima cosa da fare con una cartuccia nuova è formattarla; questa operazione serve a predisporne lo spazio in maniera adatta a ricevere i dati. L'istruzione per questo è FORMAT "m";1;"nome", a nostra scelta; durante questa operazione, che dura una trentina di secondi, tutto il nastro viene controllato alla ricerca di eventuali punti danneggiati, che saranno contrassegnati in modo da non essere mai utilizzati. Almeno un punto inutilizzabile c'é sempre, ed è la zona di giunzione delle estremità del nastro. Al termine compare sullo schermo l'ammontare di spazio disponibile nella cartuccia formattata; questo può variare da un massimo di oltre 100K ad un minimo garantito di 85K. Non ci è capitato di vedere né l'una né l'altra situazione; la media delle cartucce si aggira infatti intorno ai 90-95K. Se dovesse capitarne una con quantità sensibilmente inferiori di spazio, provate a formattarla di nuovo: potrebbe darsi (ci è successo) che una zona rovinata sia caduta a cavallo di due settori (il nastro viene diviso dalla formattazione in settori, ciascuno della capacità di 512 byte), rendendoli entrambi inutilizzabili; un nuovo processo può in questo caso migliorarne (anche di 2K) le cose.
L'istruzione CAT 1 (o 2,3... a seconda del Microdrive che vogliamo indicare) serve invece ad ottenere il catalogo della cartuccia inserita, cioé l'elenco, in ordine alfabetico (fantastico!) dei file, di tutti i tipi, presenti in quel momento; in coda viene indicato il numero di byte ancora liberi, espressi in K.
Tutti i comandi cui eravamo abituati con il registratore esistono anche per il Microdrive, avendo cura di interporre fra il comando e il nome del file (che qui è sempre obbligatorio, non solo durante SAVE) la sequenza *"m";1; ad indicare che l'operazione riguarda l'interfaccia (*), che il destinatario è un Microdrive ("m"), e che si tratta del numero 1 (1); quest'ultimo va indicato

OPEN



Cosa cambia con i Microdrive?

anche se si possiede un solo Microdrive. I vari SAVE, LOAD, con o senza CODE, DATA, e così via, hanno lo stesso uso e significato di prima; per quanto riguarda VERIFY, sebbene sia regolarmente utilizzabile, da molto tempo non lo usiamo più: infatti le probabilità che qualcosa vada storto qui sono veramente infime. Nella remota eventualità che questo succeda, esiste un raffinato messaggio di errore a proposito: Verification has failed, la verifica è fallita.
Dove la differenza c'é invece, eccome, è nella velocità: qualche secondo per programmi di una decina di K o array di più di mille elementi, un istante (davvero impressionante) per uno schermo; il tempo risparmiato, nel caso di un uso intenso, è realmente notevole.
Veniamo adesso a tutta la parte inedita dei comandi, quelli che rendono i Microdrive simili a dei floppy e lo Spectrum espanso un sistema enormemente più potente di quello inespanso. Con l'istruzione OPEN #6; "m";1;"pippo" si apre un file (il manuale lo chiama "file dati" per distinguerlo dai "file programma", ma anche gli array sono file di dati, pur essendo molto diversi da questi) di nome "pippo" e lo si associa al canale 6; in esso di potrà "scrivere" ciò che si vuole, numeri o caratteri, o tutte e due le cose insieme. con l'istruzione PRINT #6; seguita da variabili numeriche, variabili stringa, numeri o stringhe. Terminata la scrittura il file verrà chiuso con CLOSE #6. Riaprendolo successivamente, con l'istruzione uguale a quella vista prima, non sarà più permesso scrivere (provateci, e vedrete che messaggio...) ma si potrà leggere ciò che era stato immesso precedentemente, con INPUT #5;... seguito dalla o dalle variabili, che dovranno concordare per tipo (numeriche o stringhe) con i dati che devono ricevere. È ovviamente possibile tenere aperti contemporaneamente più file, alcuni in scrittura (cioè appena creati), altri in lettura, e passare con INPUT # e PRINT # dei dati dell'uno e all'altro.
Volendo fare "piazza pulita" di tutti i file rimasti aperti, si può usare CLEAR #, ma solo se si tratta di file in lettura. Usato per file in scrittura ne distruggerebbe la parte di dati non ancora inviata al Microdrive; l'invio di dati, infatti, avviene solamente quando il loro volume ha raggiunto i 512 byte (cioé la capacità di un settore), o alla chiusura del file stesso. Con questo tipo di file è possibile anche un altro tipo di operazione, che potremmo definire "lettura automatica totale". Consiste nella trascrizione dell'intero file in un altro; l'istruzione relativa è MOVE "m";1;"pippo" TO "m";1;"carlo" nel caso nessuno dei due file sia stato aperto (in questo caso apertura successiva e successiva chiusura avvengono automaticamente), o MOVE #4 TO #6 per file aperti e associati quindi a dei canali. Come molti di voi sapranno, lo Spectrum vede anche schermo, tastiera e stampante come file, associati dal sistema ai canali 2, 1 e 3, rispettivamente; questo consente di far stampare il contenuto di un file su schermo o stampante semplicemente con MOVE "m";1;"pippo" TO #2 (oppure #3), cosa estremamente utile.

A cosa possono servire

Voremmo a questo punto spendere due parole per chiarire la differenza che passa fra l'uso di file "array" e l'uso di file dati. Supponiamo di voler memorizzare un elenco di nomi e numeri telefonici, per richiamarli all'occorrenza. Dovendo usare un file "array" dovremmo avere un elenco completo dei nomi in memoria, in un insieme di caratteri (o array stringa che dir si voglia), e quindi memorizzarlo; per i numeri telefonici abbiamo due possibilità: memorizzarli come numeri con un separato insieme numerico, o trattarli come caratteri e metterli nello stesso insieme dei nomi. In ogni caso, comunque, dovremmo aver tutto il nostro archivio contemporaneamente in memoria, e la stessa cosa succederà al momento di richiamarlo per utilizzarlo. Questa situazione comporta serie limitazioni, dovute alla possibilità di memoria, nel numero massimo di nominativi del nostro elenco; tenete conto che dovrà esserci necessariamente anche un programma, in memoria, perché l'archivio possa servire a qualcosa, e questo programma occuperà anche lui la sua fetta di spazio.
Usando un file dati, invece, potremmo memorizzare ogni singolo nominativo, inviando con PRINT # la relativa stringa, seguito dal suo numero telefonico, inviando sempre con PRINT # la variabile numerica che lo contiene. In questo modo non avremo mai in memoria più di un nome ed un numero. Per recuperare un determinato nominativo, poi, basterà leggere dal file, uno alla volta, i vari nomi, usando sempe la stessa variabile per non accumularli in memoria, fino a trovare ciò che ci interessa, Volendo invece l'elenco completo dei nominativi in archivio, magari su stampante, basterà usare MOVE, senza bisogno di impiegare memoria. Limiti all'ampiezza dell'archivio: nessuno, o meglio la capienza di una cartuccia, disponendo di un Microdrive; vi sfido comunque a riempire 90K con un elenco di nomi e indirizzi...
Da questo esempio emerge come i Microdrive possano essere particolarmente utili ai possessori di uno Spectrum 16K, che sentono più forti le limitazioni di memoria. C'è però un particolare cui occorre prestare attenzione; per ciascun file aperto occorre in memoria uno spazio pari a circa 600 byte per il buffer (area di accumulo dei dati in attesa di essere inviati), la mappa e le nuove variabili di sistema. Fate quindi bene i vostri conti, perché di memoria libera nel 16K non c'è né molta (10K scarsi) e i Microdrive sono fantastici quando hanno lo spazio sufficiente per funzionare, ma possono creare un'infinità di problemi (li abbiamo sperimentati simulando una memoria piena con l'abbassamento della RAMTOP) se subiscono costrizioni.

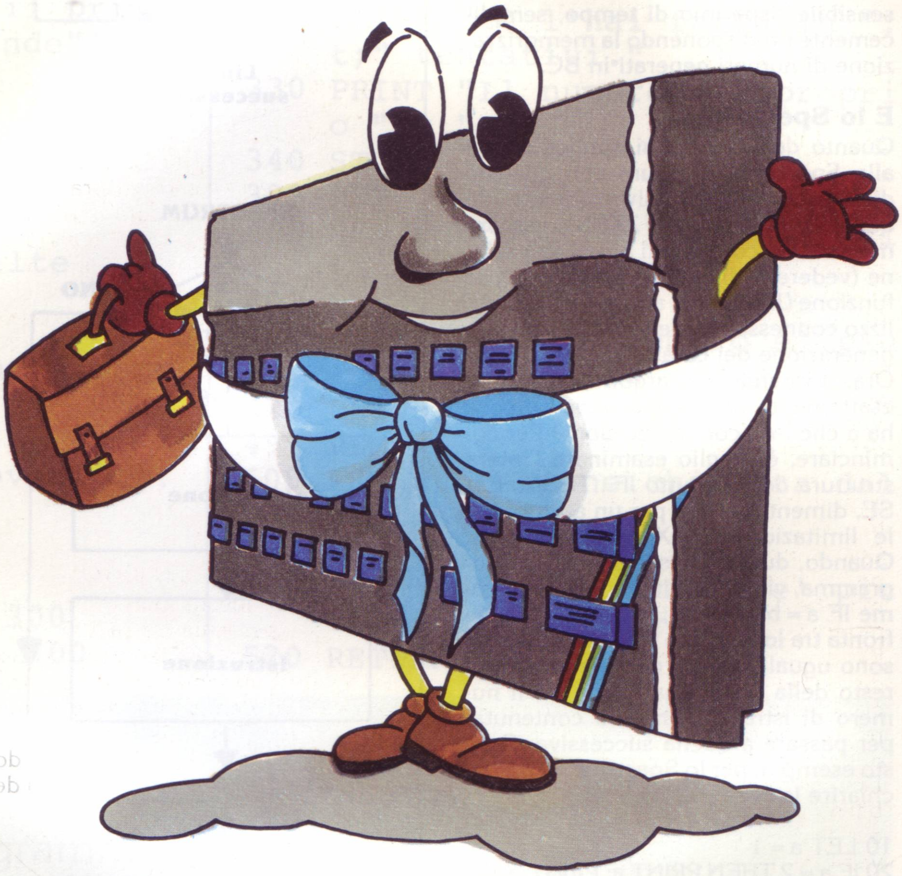
Conclusione

Il salto qualitativo che lo Spectrum compie con l'adozione dei Microdrive è indubbiamente notevole; si passa infatti da un suo uso forzatamente "giocoso" alla possibilità di renderlo utile per tanti impieghi "seri".
Purtroppo l'impostazione data al mercato italiano degli "home", propagandati praticamente come videogiochi non ha portato gradualmente l'utilizzatore a sentire la necessità di cimentarsi in usi più sofisticati e costruttivi delle loro macchine come è avvenuto all'estero (Britannia docet). Il prezzo delle cartucce, poi, non è che aiuti...

IF ...then

Decido, quindi sono... un computer

di **David Janda**
trad. e adatt. di **Lucio Bragagnolo**



Come i calcolatori (e i computer Sinclair in particolare) eseguono test per poi prendere una decisione in base al risultato.

Cos'è un linguaggio di programmazione? O meglio, che cosa qualifica un linguaggio di programmazione come tale? La risposta è: la sua capacità di compiere dei test e successivamente compiere azioni strettamente legate all'esito degli stessi. Da qui partono le fondamenta di tutti i linguaggi di programmazione, e questo è il punto di inizio a partire dal quale è possibile programmare i calcolatori in modo tale che essi possano avere un comportamento flessibile di fronte alle più svariate situazioni.

Per chiarire le cose, occorrerebbe guardare un programma BASIC sotto due punti di vista differenti - uno riguardante l'aspetto fisico del programma, e l'altro il suo aspetto logico. Il programma uno, semplicissimo ed elementare, chiede all'operatore di inserire due numeri che vengono successivamente addizionati, sottratti e moltiplicati fra di loro, dopo di che il calcolatore provvede a stampare i vari risultati.

Notate che in questo programma non vengono prese decisioni di alcun genere, e che la sequenza delle varie operazioni compiute cioé il "diagramma di flusso" segue rigorosamente un andamento "top-down", ossia dall'alto verso il basso. Ciò significa che questo programma non è molto fessibile; può infatti eseguire solamente le operazioni che noi abbiamo precedentemente programmato, utilizzando dati come i numeri che gli vengono forniti per mezzo delle istruzioni INPUT.

Costrutti di controllo

Il programma due presenta una certa differenza dal precedente. Abbiamo infatti introdotto il costrutto "IF condizione/i THEN azione/i", così che possiamo cominciare a confrontare i dati che vengono immessi, per poi effettuare determinare azioni sulla scorta dei risultati dei nostri test. Notiamo però due cose: primo, l'andamento (il flusso) del programma è ancora di tipo sequenziale, dall'alto in basso; secondo, l'istruzione IF viene usata in modo abbastanza primitivo. È possibile utilizzarla molto meglio, come si può vedere nel listato numero tre. In esso, invece di usare il costrutto IF...THEN per compiere una sola azione in seguito all'esito del test (come accadeva nel programma due), vengono usate dopo THEN istruzioni GO TO o GOSUB, che danno luogo ai cosidetti salti condizionati; con essi, il risultato del test porta all'esecuzione di un determinato gruppo di linee di programma.

I calcolatori Sinclair (con l'eccezione dell'ultimo QL) non offrono molti costrutti di controllo, per fornire unicamente l'essenziale. Altri computer, infatti (ad esempio l'Acorn BBC, diffusissimo in Gran Bretagna ma praticamente assente dalle nostre parti) lavorano su dialetti BASIC molto più dotati a questo proposito; il BASIC Sinclair è comunque sufficiente per consentirci di emulare istruzioni da esso non previste.

Per quanto riguarda lo ZX81, l'istruzionei IF ha la forma "IF condizione/i THEN azione". È possibile usarla per determinare l'esecuzione di gruppi di

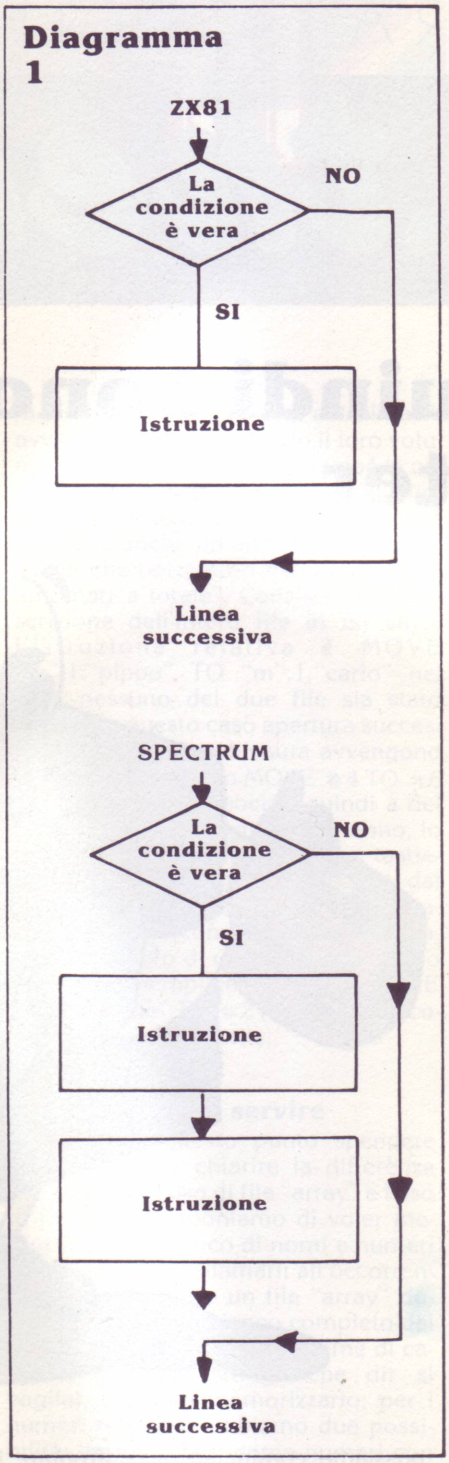
linee a piacere, "chiamandole" attraverso GOSUB o GO TO. Una terza istruzione utilizzabile è la USR, che chiama le linee desiderate attraverso il linguaggio macchina, per poi ritornare al BASIC.
Ciò è realizzabile ordinando l'esecuzione delle variabili mediante la linea LET A = USR N, dove A è la variabile numerica e N l'indirizzo di partenza del programma. Terminata l'esecuzione di quest'ultimo e restituito il controllo al BASIC, alla variabile verà assegnato il valore contenuto nella coppia di registri BC.
Può sembrare una cosa sostanzialmente inutile ma se, per esempio, disponete di una velocissima routine di generazione di numeri casuali scritta in linguaggio macchina, potete utilizzarla nella maniera sopra descritta con un sensibile risparmio di tempo, semplicemente predisponendo la memorizzazione di numeri generati in BC.

E lo Spectrum?

Quanto detto finora si applica anche allo Spectrum, sia pure con qualche differenza. Intanto, diversamente da quanto accadeva sullo ZX81, sono ammesse linee contenenti più di una azione (vedere il diagramma 1); in più, la funzione USR ha qui anche un altro utilizzo connesso, come forse sapete, alla generazione dei caratteri grafici.
Ora, tornando in argomento, cos'é esattamente una "condizione" e cosa ha a che fare con le decisioni? Per cominciare, è meglio esaminare l'intera struttura del costrutto IF...THEN...ELSE, dimenticandoci per un attimo delle limitazioni di ZX81 e Spectrum. Quando, durante l'esecuzione del programma, viene raggiunta una linea come IF a = b THEN..., il computer confronta tra loro le due variabili e, se non sono uguali, ignora completamente il resto della linea, qualunque sia il numero di istruzioni in essa contenuta, per passare a quella successiva. Questo esempio, per lo Spectrum, servirà a chiarire le cose:

```
10 LET a = 1
20 IF a = 2 THEN PRINT a: PRINT "OK"
30 PRINT "LINEA SUCCESSIVA"
```

(sullo ZX81 è sufficiente eliminare PRINT "OK"). In esecuzione, l'esito del test sarà negativo e il valore di a non verrà stampato, né lo sarà la scritta "OK", e il programma passerà alla linea seguente. In tutto questo, ripetiamo l'unica differenza tra lo Spectrum e ZX81 sta nel fatto che il primo può avere più di una istruzione dopo una IF, come è visibile nel diagramma. In realtà il computer, analizzando una condizione, non sta controllando se, per dire, A*10 = 1 o "Pinuccio" ⟨ ⟩ "Carletto"; ciò che sta facendo è una comparazione che darà come risultato "vero o falso" o, per essere più precisi, "1" o "0". Questo vale sia nel caso di un unico test come in linee più complesse come IF a = b AND c = d THEN...: per quante analisi vengano fatte il risultato finale sarà uno solo.



Dando a un esito "vero" il valore 1 e ad un esito "falso" il valore 0 possiamo ora considerare cosa accade quando abbiamo una condizione che contenga un operatore logico - AND, OR, o NOT. Prendiamo quattro variabili a, b, c, d, con valore a = 1, b = 1, c = 2, d = 3; e adesso proviamo a vedere cosa succede analizzando un delle forma IF a = b AND c = d THEN... Per prima cosa viene eseguito il test su a e b, che risulta vero (cioé 1), mentre il secondo test è falso - poiché c'è evidentemente diverso da d - e ritorna il valore 0. Siccome l'operatore logico AND richiede che per dare come risultato 1 gli operandi (nel nostro caso i due test) valgano entrambi 1, la nostra condizione è falsa, ed ha un valore 0. Ecco la "tabella della verità" completa (gli insofferenti nei confronti dell'algebra booleana trattengano il fiato per trenta secondi...) valida per l'operatore AND:

```
IF 0 AND 0 THEN 0
IF 0 AND 1 THEN 0
IF 1 AND 0 THEN 0
IF 1 AND 1 THEN 1
```

Potete vedere come il terzo elemento della tabella rappresentanti il caso che abbiamo appena considerato.
Ecco ora la tabella della verità per l'operatore OR:

```
IF 0 OR 0 THEN 0
IF 0 OR 1 THEN 1
IF 1 OR 0 THEN 1
IF 1 OR 1 THEN 1
```

L'operatore NOT non fa altro che invertire l'esito del nostro test: NOT 1 è uguale a 0, e NOT 0 è uguale a 1 (ovvio).
Per concludere, diciamo che utilizzando gli operatori logici disponibili sui calcolatori Sinclair è possibile eseguire test su ogni tipo di dati. Con un po' di allenamento, imparerete anche ad eseguire test multipli su di una sola linea, risparmiando discrete quantità di memoria e salendo un altro gradino sulla scala in cima alla quale si trovano i buoni programmatori.

Programma 1

```
10 REM programma con logica to
   p-down
20 PRINT "inserisci un numero"
30 INPUT a
40 PRINT "inseriscine un altro
   "
50 INPUT b
60 PRINT a;"+";b;"=";a+b
70 PRINT a;"-";b;"=";a-b
80 PRINT a;"*";b;"=";a*b
90 STOP
```

Programma 2

```
10 REM programma top-down con
   decisioni
20 PRINT "inserisci il primo n
   umero"
30 INPUT a
40 PRINT "e adesso inserisci i
   l secondo"
50 INPUT b
60 IF a=b THEN  PRINT "questi
   numeri sono uguali!"
70 IF a<b THEN  PRINT "il prim
   o numero e' piu' piccolo"'"
   del secondo"
80 IF a>b THEN  PRINT "il prim
   o numero e' piu' grande"'"d
   el secondo"
90 STOP
```

Programma 3

```
 10 REM decisioni e scelte
 20 LET r=INT (RND*100)
 30 LET t=0
 40 PRINT "Ho pensato un numero
    "
 50 PRINT "tra 0 e 100."
 60 PRINT "Prova a indovinarlo!
    "
 70 PRINT
 80 GO SUB 200
 90 IF g=r THEN  GO TO 300
100 IF g<r THEN  GO SUB 400
110 IF g>r THEN  GO SUB 500
120 GO TO 80
199 REM
200 REM subroutine chiamata dal
    la linea 80
201 REM
210 LET t=t+1
220 PRINT "Dimmi il numero..."
230 INPUT g
240 RETURN
299 REM
300 REM subroutine chiamata dal
    la linea 90
301 REM
310 PRINT "Bravissimo!"
320 PRINT "ce l'hai fatta in ";
    t;" tentativi."
330 PRINT "Il numero era propri
    o ";g;"!"
340 STOP
399 REM
400 REM subroutine chiamata dal
    la linea 100
401 REM
410 PRINT "Il mio numero e' piu
    ' grande di"';g
420 RETURN
499 REM
500 REM subroutine chiamata dal
    la linea 110
501 REM
510 PRINT "Il mio numero e' piu
    ' piccolo di"';g
520 RETURN
```

I programmi "Forzaquattro" e "Goldrace", entrambi per ZX81, pubblicati sul numero 3 di SUPERSINC non sono stati realizzati da Marcello Spero, come erroneamente riportato, bensì da Giovanni Tisi.
Ci scusiamo dell'errore con entrambi gli interessati.

Hardware

di **Marcello Spero**

L'interfaccia come e perché



Teoria e pratica della costruzione di alcuni dispositivi per lo sfruttamento ottimale del calcolatore

La scorsa puntata di "Nei meandri dello Spectrum" abbiamo esaminato i vari componenti dello Spectrum, con particolare riguardo alla CPU ed alle sue funzioni; abbiamo anche visto come sul retro del computer siano accessibili molti dei segnali che scandiscono la "vita" della macchina.

Questa volta esaminiamo più in dettaglio quali sono i modi e i mezzi di cui lo Spectrum dispone per collegarsi e comunicare con il mondo esterno, e perché può essere interessante sfruttarli per metterlo in collegamento con congegni di vario tipo.

Il mezzo di comunicazione con l'esterno dello Spectrum, come di qualsiasi altro computer, consiste in dati, cioè numeri, per intenderci. Numeri che, in forma binaria, entrano ed escono attraverso otto linee, che prendono il nome di "bus dei dati". Le linee sono otto perché questo è il numero di cifre binarie, cioè di bit, che forma un byte, l'unità di informazione della CPU dello Spectrum.

Sul bus dei dati, però, si muovono anche le informazioni destinate alla memoria o provenienti da essa; per distinguere i due tipi di destinatario o di "mittente" ci sono due linee apposite, attivate dalla CPU, l'una per indicare che l'informazione riguarda la memoria, l'altra per indicare che riguarda il mondo esterno, chiamato I/O (Input/Output). Rimane da determinare se l'informazione proviene dalla CPU o vi è diretta; per questo ci sono altre due linee, l'una per indicare che la CPU vuole leggere il dato, l'altra per indicare che la CPU vuole scrivere il dato, che quindi proverrà da lei.

Avendo fatto sapere con chi vuole avere a che fare, e se vuole inviare o ricevere dati, la CPU deve anche chiarire quale locazione vuole nel caso della memoria, o quale periferica nel caso dell'I/O. A questo è deputato un altro bus, il "bus degli indirizzi", che verrà caricato con un valore contemporanea-

mente al bus dei dati. Questo valore rappresenta l'indirizzo, sempre in forma binaria ma stavolta a 16 cifre cioè a due byte, della locazione di memoria o della periferica richiesta.
Rivediamo il tutto dal punto di vista di un congegno esterno che voglia collegarsi allo Spectrum e debba quindi capire quando l'informazione sul bus dei dati è diretta a lui. La condizione che lo interessa sarà verificata se saranno attivate (cioè portate a livello logico zero) contemporaneamente la linea I/0 (si chiama $\overline{\text{I/ORQ}}$) e quella di scrittura (si chiama $\overline{\text{WR}}$), e se sul bus degli indirizzi ci sarà il suo numero. Per inviare un dato dovrà invece attendere che siano attivate $\overline{\text{I/ORQ}}$ e la linea di lettura ($\overline{\text{RD}}$), e che il bus degli indirizzi abbia il suo numero.
Un congegno che sappia leggere tutte queste linee, prenda in considerazione solo le informazioni che lo riguardano e attenda che il suo momento prima di inviarne è un "interfaccia".
Un certo numero di periferiche essenziali è già collegato con lo Spectrum: si tratta della tastiera, del registratore e dell'altoparlante; altre, come la stampante ZX o i Microdrive, possono essergli collegate. Queste però possiedono già una loro interfaccia incorporata, che permette loro di funzionare senza problemi. Quello che noi ci proponiamo, invece, è la realizzazione di un'interfaccia che consenta il collegamento a congegni non originariamente concepiti per funzionare in unione con lo Spectrum. Per avere chiaro il motivo che rende attraente un simile lavoro, vediamo un esempio.
Supponiamo di voler dotare la porta di casa nostra di una serratura a combinazione, formata da una pulsantiera esterna, collegata ad un apposito congegno interno che fa scattare la serratura solo se è stata composta la combinazione corretta. Oltre ad una pulsantiera ed una serratura elettrica, ci occorre un apposito circuito che riceva i segnali elettrici provenienti dai tasti, ne controlli la combinazione ed eventualmente dia tensione alla serratura.
Volendo realizzare qualcosa di veramente sicuro, però, bisognerebbe che questo circuito, pur non reagendo ad una prima combinazione sbagliata, dopo un determinato numero di tentativi infruttuosi non ne accettasse più altre per un certo tempo, così da scoraggiare eventuali malintenzionati. Per questo dovrebbe essere dotato di un temporizzatore, che parta alla prima combinazione sbagliata.
Fatti i conti ci si rende conto che la spesa per un simile apparecchio non sarebbe indifferente; oltretutto se un giorno decidessimo che questo tipo di serratura non è più adatto alla nostra casa dovremmo buttare tutto, perché un congegno così specifico ben difficilmente potrà essere adattato ad un altro scopo.
Supponiamo invece di realizzare la medesima serratura servendoci dello Spectrum. Oltre ovviamente, alla pulsantiera ed alla serratura, non ci serve nient'altro: ci basta il computer e la sua interfaccia. Un apposito programma, infatti, leggerà lo stato della pulsantiera, lo paragonerà alla combinazione esatta ed eventualmente invierà un segnale che farà scattare la serratura.
La protezione a tempo vista prima diventa di realizzazione semplicissima senza alcuna aggiunta di materiale; sempre senza aggiunta di materiale sono possibili molti servizi aggiuntivi, quali una combinazione personalizzata per ciascun membro della famiglia con relativa registrazione delle entrate, o combinazioni che aprono la porta solo entro orari determinati. Altre aggiunte o modifiche sarebbero sempre possibili, mentre nel caso di una eliminazione di tutto il sistema l'unica cosa da buttare (non è detto) sarebbe la pulsantiera.
Il nostro scopo non è certamente la realizzazione di serrature a combinazione, ma quello che l'esempio voleva mettere in luce è che utilizzando un computer in un sistema di controllo esterno si realizza un notevole risparmio, ed una grande versatilità d'uso. In pratica un programma fa le veci di un circuito, cioè il software consente di risparmiare sull'hardware.
Si apre così la strada ad una moltitudine di congegni, dai più utili ai più inutili: dal combinatore telefonico che, ricevuto da tastiera il nome e cognome di una persona, ne trova il numero e lo compone, ripetendolo finché non lo trova libero, all'antifurto più sofisticato ed imbattibile, all'apparecchio che al mattino accende la caffettiera elettrica e ci dice sullo schermo se pioverà o meno, le possibilità sono infinite.
Esiste poi tutta una gamma di informazioni che non sono espresse come valori numerici distinti, cioè acceso o spento, uno o zero, dieci o undici, ma con quantità variabili in modo continuo. Per intenderci, pensate al livello dell'acqua in una vasca che si riempie, o alla temperatura. In questi casi per rendere le informazioni accessibili al computer occorre trasformarle in una successione di valori discreti, necessariamente approssimata alla variazione continua. L'apparecchio che opera questa trasformazione si chiama convertitore analogico/digitale, abbreviato in A/D; analogico è infatti un segnale che può variare in modo continuo, mentre è digitale un segnale che può assumere solo un certo numero di valori discreti.
Con un apparecchio di questo tipo, che arriveremo a fabbricarci, avremo possibilità ancora diverse: misurazione di temperature, voltmetri, penne ottiche, e così via.
Ma torniamo con i piedi per terra, e vediamo quali sono i componenti che ci servono per realizzare una semplice interfaccia. La nostra prima realizzazione sarà improntata alla masssima semplicità; avrà la possibilità di collegarsi ad un solo congegno esterno, e quindi dovrà riconoscere un solo indirizzo, che per semplicità sarà costituito da un solo bit. Quando quel bit sarà attivato, cioè avrà valore logico zero, il "port" cioè la via di comunicazione, da noi creato sarà selezionato. Naturalmente dovranno essere lette contemporaneamente tutte le linee necessarie, cioè $\overline{\text{I/O}}$, $\overline{\text{RD}}$ e $\overline{\text{WR}}$.
In pratica abbiamo bisogno di ricavare quattro segnali in ingresso due segnali in uscita, che diano le informazioni, che chiameremo OKINPUT e OKOUTPUT, per attivare l'una o l'altra sezione dell'interfaccia. Questo risultato lo si ottiene usando delle "porte nor", cioè dei circuiti che eseguono l'operaziono logica "or" sui segnali in ingresso, e quindi l'operazione logica "not" sul risultato ottenuto. In tabella 1 vedete la cosiddetta "tavola di verità" della funzione nor a tre ingressi, che è appunto quella che ci interessa. In pratica collegheremo le varie linee alle due porte logiche in modo che OKINPUT venga attivato (in questo caso l'attivazione corrisponde al livello logico 1) quando $\overline{\text{I/O}}$, il bit del bus indirizzi e $\overline{\text{RD}}$ sono tutti tre a livello logico zero; OKOUTPUT sarà invece attivato quando sono a zero $\overline{\text{I/O}}$, il bit del bus indirizzi e $\overline{\text{WR}}$. OKINPUT abiliterà un apposito circuito a trasmettere il dato che aveva precedentemente immagazzinato, mentre OKOUTPUT ne abiliterà un altro a "lasciar passare" l'informazione proveniente dal bus dati.
Abbiamo così tracciato lo schema della nostra interfaccia. La prossima volta vedremo la parte pratica del progetto, ossia la realizzazione vera e propria, e vi accorgerete che queste cose sono ben più complicate a dirsi che a farsi.

TAVOLA DI VERITÀ DELLA FUNZIONE LOGICA NOR A TRE INGRESSI:

INGRESSI			USCITA
0	0	0	1
1	0	0	0
0	1	0	0
1	1	0	0
0	0	1	0
1	0	1	0
0	1	1	0
1	1	1	0

tabella 1

ERRATA CORRIGE

Il labirinto di "Anellide"

Il listato del gioco "Anellide" comparso su SUPERSINC numero 3 a pagina 48 è stato pubblicato privo di dati necessari per la rappresentazione del labirinto sullo schermo. È possibile rimediare all'inconveniente in vari sistemi: 1) cancellate la linea 9999 del listato originale e salvatelo con GO TO 9998.

Ora date pure NEW e scrivete un programmino caricatore, che servirà a inserire nelle locazioni da 31000 a 31639 (quelle esaminate dal programma per mezzo delle linee 80/83) i codici dei caratteri occorrenti per disegnare il labirinto. Un esempio può essere

```
10 FOR i=31000 TO 31639
20 INPUT x
30 IF x<> 32 AND (x<144 OR x>164) THEN BEEP 1,1: PRINT "Hai sbagliato!": GO TO 20: REM "sicura" per limitare la possibilità di errori
40 PRINT i,x
50 POKE i,x,
60 NEXT i
```

Date il RUN e inserite, con somma pazienza, i codici (che potete leggere in questa pagina). Arrivati alla fine... vi conviene controllare tutto utilizzando lo stesso programmino, privato delle linee 20,30,50 e con la linea 40 modificata in 40 PRINT i, PEEK i. Se volete evitare la domandina "scroll?", potete inserire una linea 45 INPUT INKEY$ = "y" o, meglio ancora, 45 POKE 23692,255. Terminato? Bene! Non vi vresta che salvare il tutto di seguito al listato originale, mediante un SAVE "labirinto" CODE 31000,640. Digitate RAND USR 0 e caricate "Anellide", ora perfettamente funzionante. 2) Inserite gli stessi codici, ma direttamente nel corpo del programma, sotto forma di linee DATA, modificando il listato come segue:

```
80 FOR n=1 TO 640
81 READ c
83 PRINT INK 3; CHR$ c;: NEXT N
85 DATA...
86 DATA...
87 DATA...
```

e così via. In questo caso, dovete cancellare la linea 9999 e il LOAD " " CODE della linea 1, per poi salvare il gioco col solito GO TO 9998. 3) Infine è possibile ricavare il labirinto dalla foto pubblicata nell'articolo individuando i caratteri grafici utilizzati per poi creare alcune stringhe (contenenti ognuna una o più righe del labirinto) da inserire nel programma tra la linea 70 e la linea 96, accompagnate da istruzioni PRINT. Le linee da eliminare sono queste: 80,81,83,9999 (più il LOAD " " CODE di linea 1) e la procedura di registrazione consiste nel solito GO TO 9998. Sperando di avere risolto con questa spiegazione ogni problema, ci scusiamo con tutti i lettori per questo spiacevole incidente.

Anellide
versione per ZX Spectrum 16K/48K
lista dei codici contenuti nelle locazioni 31000/31639

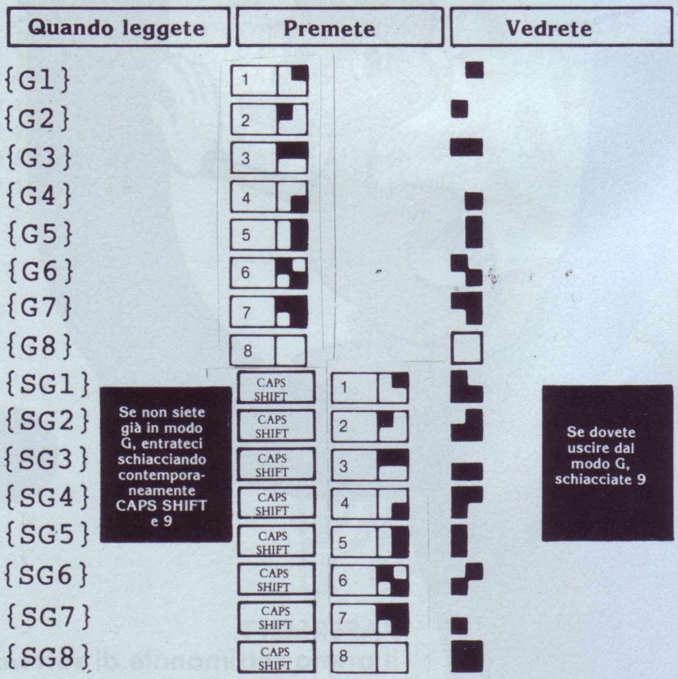
31000	150	31037	32	31074	32	31117	149
31001	149	31038	148	31075	32	31118	157
31002	149	31039	32	31076	144	31119	32
31003	149	31040	32	31077	32	31120	145
31004	149	31041	32	31078	148	31121	32
31005	149	31042	145	31079	32	31122	148
31006	154	31043	32	31080	144	31123	32
31007	149	31044	32	31081	32	31124	32
31008	149	31045	32	31082	148	31125	32
31009	149	31046	148	31083	32	31126	32
31010	154	31047	32	31084	32	31127	148
31011	149	31048	32	31085	32	31128	156
31012	149	31049	32	31086	144	31129	146
31013	149	31050	148	31087	32	31130	32
31014	154	31051	32	31088	148	31131	32
31015	149	31052	145	31089	32	31132	32
31016	149	31053	32	31090	150	31133	147
31017	149	31054	32	31091	149	31134	152
31018	151	31055	32	31092	149	31135	32
31019	32	31056	148	31093	146	31136	32
31020	150	31057	32	31094	32	31137	32
31021	149	31058	32	31095	148	31138	32
31022	149	31059	32	31096	148	31139	32
31023	149	31060	32	31097	32	31140	32
31024	154	31061	32	31098	32	31141	32
31025	149	31062	32	31099	145	31142	148
31026	149	31063	148	31100	32	31143	32
31027	149	31064	148	31101	32	31144	32
31028	149	31065	32	31102	148	31145	32
31029	149	31066	147	31103	32	31146	32
31030	149	31067	154	31104	153	31147	32
31031	151	31068	146	31105	149	31148	32
31032	148	31069	32	31106	149	31149	32
31033	32	31070	148	31107	149	31150	148
31034	32	31071	32	31108	152	31151	32
31035	32	31072	144	31109	32	31152	32
31036	32	31073	32	31110	148	31153	32
				31111	32	31154	148
				31112	145	31155	32
				31113	32	31156	144
				31114	145	31157	32
				31115	32	31158	150
				31116	147	31159	157

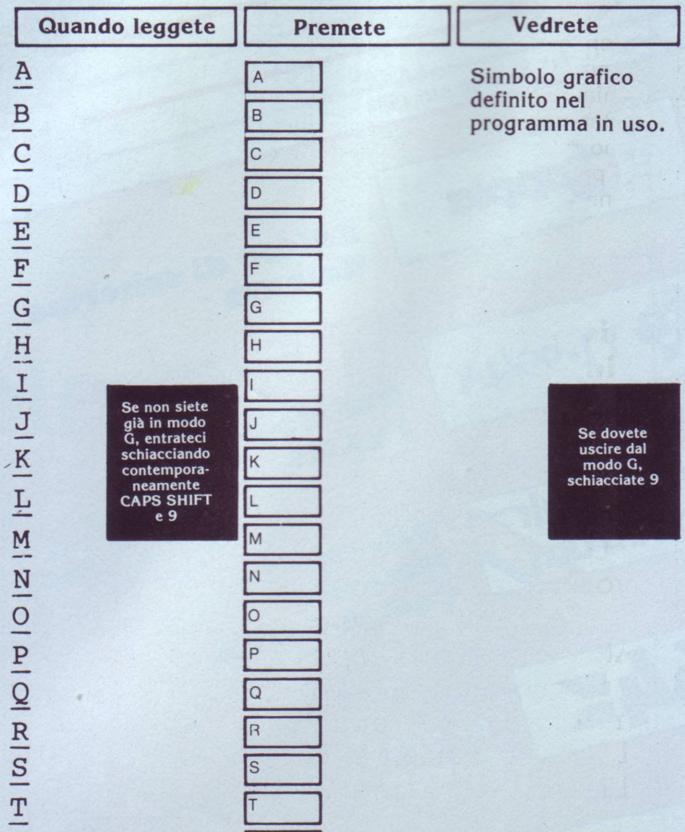
31160	148	31231	32	31302	149	31373	146
31161	32	31232	32	31303	149	31374	32
31162	32	31233	32	31304	146	31375	145
31163	144	31234	148	31305	32	31376	32
31164	32	31235	32	31306	148	31377	32
31165	32	31236	153	31307	32	31378	32
31166	32	31237	149	31308	144	31379	145
31167	32	31238	146	31309	32	31380	32
31168	144	31239	32	31310	147	31381	148
31169	32	31240	147	31311	151	31382	32
31170	144	31241	149	31312	32	31383	145
31171	32	31242	151	31313	32	31384	32
31172	144	31243	32	31314	32	31385	32
31173	32	31244	145	31315	32	31386	32
31174	145	31245	32	31316	32	31387	32
31175	32	31246	147	31317	144	31388	32
31176	147	31247	149	31318	32	31389	32
31177	149	31248	155	31319	148	31390	153
31178	146	31249	151	31320	148	31391	149
31179	32	31250	32	31321	32	31392	151
31180	144	31251	150	31322	144	31393	32
31181	32	31252	149	31323	32	31394	32
31182	145	31253	146	31324	32	31395	32
31183	32	31254	32	31325	32	31396	32
31184	144	31255	148	31326	148	31397	32
31185	32	31256	156	31327	32	31398	148
31186	145	31257	149	31328	147	31399	32
31187	32	31258	149	31329	149	31400	148
31188	145	31259	149	31330	151	31401	32
31189	32	31260	151	31331	32	31402	32
31190	153	31261	32	31332	148	31403	32
31191	157	31262	150	31333	32	31404	32
31192	148	31263	146	31334	32	31405	32
31193	32	31264	32	31335	32	31406	32
31194	147	31265	147	31336	32	31407	32
31195	155	31266	152	31337	32	31408	32
31196	149	31267	32	31338	148	31409	144
31197	149	31268	32	31339	32	31410	32
31198	149	31269	32	31340	148	31411	32
31199	149	31270	32	31341	32	31412	32
31200	152	31271	32	31342	32	31413	148
31201	32	31272	32	31343	156	31414	32
31202	148	31273	32	31344	149	31415	32
31203	32	31274	148	31345	149	31416	144
31204	148	31275	32	31346	149	31417	32
31205	32	31276	32	31347	151	31418	144
31206	32	31277	32	31348	32	31419	32
31207	32	31278	32	31349	148	31420	144
31208	32	31279	32	31350	32	31421	32
31209	32	31280	32	31351	148	31422	32
31210	32	31281	145	31352	145	31423	32
31211	32	31282	32	31353	32	31424	148
31212	148	31283	145	31354	153	31425	32
31213	32	31284	32	31355	149	31426	147
31214	32	31285	32	31356	146	31427	149
31215	32	31286	32	31357	32	31428	151
31216	148	31287	148	31358	148	31429	32
31217	32	31288	148	31359	32	31430	145
31218	32	31289	32	31360	32	31431	32
31219	32	31290	32	31361	32	31432	145
31220	32	31291	32	31362	145	31433	32
31221	32	31292	145	31363	32	31434	147
31222	32	31293	32	31364	145	31435	149
31223	148	31294	148	31365	32	31436	149
31224	148	31295	32	31366	144	31437	151
31225	32	31296	32	31367	32	31438	32
31226	32	31297	32	31368	144	31439	144
31227	32	31298	32	31369	32	31440	32
31228	32	31299	32	31370	145	31441	156
31229	32	31300	150	31371	32	31442	149
31230	32	31301	149	31372	153	31443	146
						31444	32

31445	145	31494	144	31543	148	31592	32
31446	32	31495	32	31544	148	31593	32
31447	144	31496	150	31545	32	31594	32
31448	148	31497	146	31546	145	31595	32
31449	32	31498	32	31547	32	31596	32
31450	148	31499	144	31548	145	31597	148
31451	32	31500	32	31549	32	31598	32
31452	153	31501	32	31550	148	31599	32
31453	149	31502	32	31551	32	31600	32
31454	146	31503	32	31552	145	31601	148
31455	32	31504	32	31553	32	31602	32
31456	148	31505	145	31554	144	31603	32
31457	32	31506	32	31555	32	31604	32
31458	32	31507	147	31556	147	31605	32
31459	32	31508	149	31557	149	31606	32
31460	148	31509	146	31558	152	31607	148
31461	32	31510	32	31559	32	31608	153
31462	32	31511	148	31560	145	31609	149
31463	32	31512	148	31561	32	31610	149
31464	32	31513	32	31562	147	31611	149
31465	32	31514	148	31563	152	31612	149
31466	32	31515	32	31564	32	31613	149
31467	32	31516	144	31565	148	31614	155
31468	32	31517	32	31566	32	31615	149
31469	145	31518	144	31567	145	31616	149
31470	32	31519	32	31568	32	31617	149
31471	145	31520	148	31569	144	31618	155
31472	32	31521	32	31570	32	31619	149
31473	148	31522	32	31571	147	31620	149
31474	32	31523	32	31572	149	31621	149
31475	32	31524	32	31573	146	31622	149
31476	32	31525	32	31574	32	31623	149
31477	32	31526	148	31575	148	31624	149
31478	32	31527	32	31576	148	31625	149
31479	148	31528	148	31577	32	31626	146
31480	148	31529	32	31578	32	31627	32
31481	32	31530	32	31579	32	31628	147
31482	148	31531	148	31580	32	31629	155
31483	32	31532	32	31581	32	31630	149
31484	32	31533	144	31582	148	31631	149
31485	32	31534	32	31583	32	31632	149
31486	32	31535	144	31584	32	31633	155
31487	32	31536	32	31585	32	31634	149
31488	156	31537	32	31586	148	31635	149
31489	149	31538	32	31587	32	31636	149
31490	146	31539	32	31588	32	31637	149
31491	32	31540	32	31589	32	31638	149
31492	145	31541	32	31590	32	31639	152
31493	32	31542	32	31591	32		

Piccola guida all'input dei programmi

I programmi pubblicati da SUPERSINC sono stati accuratamente provati e verificati. In questo modo speriamo di avere ridotto al minimo la possibilità di errori.
Nel caso comunque un programma dovesse presentare malfunzionamenti, pubblicheremo le modifiche da apportare in uno dei numeri immediatamente successivi della rivista.
Per utilizzare i programmi pubblicati è sufficiente digitare i relativi listati sulla tastiera del calcolatore.
Nel caso dello ZX SPECTRUM, abbiamo previsto una semplice decodifica dei tasti da digitare nel modo grafico, per semplificare la comprensione dei listati.
Come è noto, lo SPECTRUM è provvisto di 2 serie di tasti grafici: una prima serie di 16 caratteri grafici predefiniti (i tasti numerici da 1 a 8 e gli stessi "shiftati") e una serie di caratteri definibili dall'utente nell'ambito di un programma (le lettere da A a U).
In entrambi i casi, per ottenere i caratteri desiderati occorre entrare in modo grafico (cursore contrassegnato dalla lettera G lampeggiante) premendo contemporaneamente i tasti CAPS-SHIFT e 9.
Nei nostri listati i caratteri grafici predefiniti sono indicati da una G seguita dal numero corrispondente al testo che occorre digitare, il tutto racchiuso tra due parentesi graffe.
Ad esempio {G4} significa che occorre digitare il tasto 4, con il cursore in modo grafico.
Analogamente la codifica SG, seguita da un numero da 1 a 8, significa che occorre digitare il relativo tasto numerico premendo contemporaneamente il tasto CAPS-SHIFT.
Ad esempio quando si trova la codifica {SG2}, occorre premere il tasto 2 contemporaneamente al tasto CAPS-SHIFT, ovviamente con il cursore in modo grafico. In entrambi i casi precedenti, quando un simbolo grafico deve essere digitato più volte, i caratteri G o SG della codifica sono preceduti da un numero che specifica quante volte va premuto il tasto grafico indicato.
Così ad esempio {8G5} significa che il tasto grafico 5 va digitato 8 volte e analogamente {4SG1} indica che il tasto grafico 1 e CAPS-SHIFT devono essere battuti 4 volte.
I caratteri grafici definibili (le lettere da A a U in modo grafico) hanno una codifica semplificata: la lettera corrispondente, sottolineata.
Quando in un listato viene presentata, ad esempio, una A sottolineata, occorre entrare in modo grafico (al solito premendo contemporaneamente i tasti CAPS-SHIFT e 9) e quindi digitare semplicemente il tasto che contrassegna la lettera A.





Archivio

di **Enrico Levantino**

Un piccolo ma efficiente data base a vostra disposizione

Questo programma gestisce una base di dati, cioè consente una raccolta di informazioni relative a una data applicazione (rubriche telefoniche, biblioteca di casa, elenco dei programmi, ecc.). Ogni informazione è racchiusa in un record (un numero telefonico, un libro, un programma..) composto da campi, ovvero informazioni più specifiche sul soggetto in esame. Nell'archivio della biblioteca domestica ci saranno così diversi record ognuno contenente un libro; in tutti i record ci sarà la medesima strutturazione in campi (titolo del libro, autore, editore, prezzo, data di acquisto, eventuale prestito, numero di pagine e così via). L'archivio può essere aggiornato modificando, aggiungendo o togliendo record; è possibile poi rintracciare un record indicandone il contenuto di un campo.
Non spendiamo altre parole, poiché il programma già di per sé stesso abbonda in chiarimenti e spiegazioni. Quanto al listato, le REM in esso contenute illustrano i compiti svolti dalle varie routine presenti, favorendone la comprensione.



Archivio
versione per ZX Spectrum 16K/48K

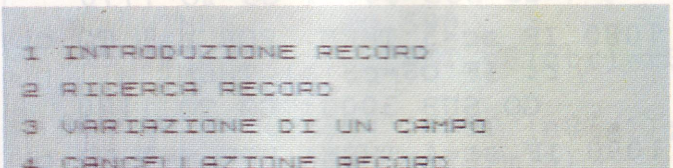
```
30 PRINT "Questo programma con
   sente la creazione di li
   ste di record. La definiz
   ione della loro      quant
   ita' e lunghezza e' a
   vostro piacere, sempre nei
      limiti imposti dalla R
   AM a      disposizione. Per
    i meno espertidiremo che u
   n record e' una     serie d
   i informazioni omogenee  su
   llo stesso soggetto."
31 PRINT "Un record e' compost
   o da campi, ognuno dei qual
   i contiene una   informazio
   ne specifica. Esempio:in un
   a rubrica telefonica puo'
   esserci un campo chiamato N
   OME, un campo NUM.TEL. e co
   si via, per una gamma di
   applicazioni   limitata sol
   amente dalla vostra immagin
   azione."
32 PRINT #0;"premi un tasto pe
   r proseguire": PAUSE 0:
   CLS
```

```
 33 PRINT "Una volta stabilito
    il numero   dei record, i c
    ampi che li     formano e
    la lunghezza di questiultim
    i, non sara' piu' possibile
    modificare la struttura di
        quell'archivio. Eviden
    temente   e' possibile aver
    e su cassetta  diversi arch
    ivi (gestibili uno  alla vo
    lta) strutturati in modi di
    versi."'"Nel caso che intro
    duciate un  dato piu' lun
    go del consentito  il compu
    ter non lo accettera', per
    mettendovi di reinserirlo
      correttamente."
 34 PRINT #0;"premi un tasto pe
    r proseguire": PAUSE 0
 80 CLS : LET ctr=0: LET variaz
    ione=0: LET cancellazione=0
    : LET z$="ARCHIVIO"
 90 INPUT "quanti records? ";re
100 INPUT "Quanti campi vuoi pe
    r record? (massimo 10) ";ca
110 IF ca>10 THEN  GO TO 100
120 DIM n$(10,10): DIM m(10)
125 FOR a=1 TO 10: LET m(a)=1:
     NEXT a
130 FOR a=1 TO ca
140 INPUT "nome ";(a);"' campo"
    ;n$(a)
160 INPUT "lunghezza ";(a);"' c
    ampo(max32)";m(a)
170 IF m(a)>32 THEN  GO TO 160
175 NEXT a
180 DIM a$(re,m(1)): DIM b$(re,
    m(2)): DIM c$(re,m(3)):
     DIM d$(re,m(4)): DIM e$(re
    ,m(5)): DIM f$(re,m(6)):
     DIM g$(re,m(7)): DIM h$(re
    ,m(8)): DIM i$(re,m(9)):
     DIM l$(re,m(10))
190 CLS : PRINT "1 INTRODUZIONE
     RECORD": PRINT : PRINT "2
    RICERCA RECORD": PRINT :
     PRINT "3 VARIAZIONE DI UN
    CAMPO": PRINT : PRINT "4 CA
    NCELLAZIONE RECORD":
     PRINT : PRINT "5 SALVATAGG
    IO DATI"
200 INPUT "Opzione scelta?";op
210 IF op>5 THEN  GO TO 200
220 IF op=1 THEN  GO SUB 500
230 IF op=2 THEN  GO TO 1000
240 IF op=3 THEN  GO SUB 1500
250 IF op=4 THEN  GO SUB 2000
260 IF op=5 THEN  GO SUB 2500
270 GO TO 190
500 REM introduzione dati
505 CLS
510 LET ctr=ctr+1
520 FOR a=1 TO ca
530 IF a=1 THEN  INPUT (n$(a));
    " ";r$: IF LEN r$>m(a)
     THEN  GO TO 530
535 IF a=1 THEN  LET a$(ctr)=r$
540 IF a=2 THEN  INPUT (n$(a));
    " ";r$: IF LEN r$>m(a)
     THEN  GO TO 540
545 IF a=2 THEN  LET b$(ctr)=r$
550 IF a=3 THEN  INPUT (n$(a));
    " ";r$: IF LEN r$>m(a)
     THEN  GO TO 550
555 IF a=3 THEN  LET c$(ctr)=r$
560 IF a=4 THEN  INPUT (n$(a));
    " ";r$: IF LEN r$>m(a)
     THEN  GO TO 560
565 IF a=4 THEN  LET d$(ctr)=r$
570 IF a=5 THEN  INPUT (n$(a));
    " ";r$: IF LEN r$>m(a)
     THEN  GO TO 570
575 IF a=5 THEN  LET e$(ctr)=r$
580 IF a=6 THEN  INPUT (n$(a));
    " ";r$: IF LEN r$>m(a)
     THEN  GO TO 580
585 IF a=6 THEN  LET f$(ctr)=r$
590 IF a=7 THEN  INPUT (n$(a));
    " ";r$: IF LEN r$>m(a)
     THEN  GO TO 590
595 IF a=7 THEN  LET g$(ctr)=r$
600 IF a=8 THEN  INPUT (n$(a));
    " ";r$: IF LEN r$>m(a)
     THEN  GO TO 600
605 IF a=8 THEN  LET h$(ctr)=r$
610 IF a=9 THEN  INPUT (n$(a));
    " ";r$: IF LEN r$>m(a)
     THEN  GO TO 610
615 IF a=9 THEN  LET i$(ctr)=r$
620 IF a=10 THEN  INPUT (n$(a))
    ;" ";r$: IF LEN r$>m(a)
     THEN  GO TO 620
```

```
625 IF a=10 THEN  LET l$(ctr)=r
   $
630 NEXT a
640 LET a=ctr: GO SUB 3000
650 INPUT "Sono corretti i dati
    introdotti? (s/n) ";r$
660 IF r$<>"s" AND r$<>"n"
    THEN  GO TO 650
670 IF r$="n" THEN  GO TO 520
680 INPUT "Vuoi introdurre un a
   ltro Record? (s/n) ";r$
690 IF r$<>"n" AND r$<>"s"
    THEN  GO TO 680
695 IF r$="s" AND ctr=re THEN
    PRINT #0;"Non c'e" piu' sp
   azio in archivio": PRINT #1
   ;"premi un tasto per proseg
   uire": PAUSE 0: GO TO 190
700 IF r$="s" THEN  GO TO 510
710 RETURN
1000 REM ricerca record
1010 CLS
1020 FOR x=1 TO ca: PRINT x;" ";
   n$(x): PRINT : NEXT x
1030 INPUT "Campo con cui cercar
   e il record? ";sc
1040 IF sc>ca THEN  GO TO 1030
1050 INPUT "introduci il dato pe
   r la ricerca ";o$
1055 IF LEN o$>m(sc) THEN
    GO TO 1050
1058 LET r=m(sc)-LEN o$: LET o$=
   o$+"                   "(1 TO r)
1060 IF sc=1 THEN  FOR a=1 TO ct
   r: IF o$=a$(a) THEN
    GO SUB 3000: GO TO 1170
1070 IF sc=2 THEN  FOR a=1 TO ct
   r: IF o$=b$(a) THEN
    GO SUB 3000: GO TO 1170
1080 IF sc=3 THEN  FOR a=1 TO ct
   r: IF o$=c$(a) THEN
    GO SUB 3000: GO TO 1170
1090 IF sc=4 THEN  FOR a=1 TO ct
   r: IF o$=d$(a) THEN
    GO SUB 3000: GO TO 1170
1100 IF sc=5 THEN  FOR a=1 TO ct
   r: IF o$=e$(a) THEN
    GO SUB 3000: GO TO 1170
1110 IF sc=6 THEN  FOR a=1 TO ct
   r: IF o$=f$(a) THEN
    GO SUB 3000: GO TO 1170
1120 IF sc=7 THEN  FOR a=1 TO ct
   r: IF o$=g$(a) THEN
    GO SUB 3000: GO TO 1170
1130 IF sc=8 THEN  FOR a=1 TO ct
   r: IF o$=h$(a) THEN
    GO SUB 3000: GO TO 1170
1140 IF sc=9 THEN  FOR a=1 TO ct
   r: IF o$=i$(a) THEN
    GO SUB 3000: GO TO 1170
1150 IF sc=10 THEN  FOR a=1 TO c
   tr: IF o$=l$(a) THEN
    GO SUB 3000: GO TO 1170
1155 NEXT a
1160 PRINT #0;"Il dato introdott
   o non c'e:forsehai scritto
   male": PAUSE 200: GO TO 119
   0
1170 IF variazione=1 THEN  LET v
   ariazione=0: RETURN
1180 IF cancellazione=1 THEN
    LET cancellazione=0:
    RETURN
1190 INPUT "Vuoi vedere un altro
    record? (s/n) ";r$
1200 IF r$<>"s" AND r$<>"n"
    THEN  GO TO 1190
1210 IF r$="s" THEN  GO TO 1010
1215 LET variazione=0: LET cance
   llazione=0
1220 GO TO 190
1500 REM variazione campo
1505 CLS
1510 LET variazione=1:
    GO SUB 1000
1520 INPUT "Quale campo? (scrivi
    il numero corrispondente)
   ";ri
1525 IF ri>ca THEN  GO TO 1520
1530 INPUT "Con cosa vuoi sostit
   uirlo? ";s$
1540 IF ri=1 THEN  IF LEN s$>m(1
   ) THEN  GO TO 1530
1550 IF ri=1 THEN  LET a$(a)=s$:
    GO SUB 3000
1560 IF ri=2 THEN  IF LEN s$>m(2
   ) THEN  GO TO 1530
1570 IF ri=2 THEN  LET b$(a)=s$:
    GO SUB 3000
1580 IF ri=3 THEN  IF LEN s$>m(3
   ) THEN  GO TO 1530
1590 IF ri=3 THEN  LET c$(a)=s$:
    GO SUB 3000
1600 IF ri=4 THEN  IF LEN s$>m(4
   ) THEN  GO TO 1530
1610 IF ri=4 THEN  LET d$(a)=s$:
    GO SUB 3000
1620 IF ri=5 THEN  IF LEN s$>m(5
   ) THEN  GO TO 1530
1630 IF ri=5 THEN  LET e$(a)=s$:
    GO SUB 3000
```

```
1640 IF ri=6 THEN  IF LEN s$>m(6
    ) THEN  GO TO 1530
1650 IF ri=6 THEN  LET f$(a)=s$:
     GO SUB 3000
1660 IF ri=7 THEN  IF LEN s$>m(7
    ) THEN  GO TO 1530
1670 IF ri=7 THEN  LET g$(a)=s$:
     GO SUB 3000
1680 IF ri=8 THEN  IF LEN s$>m(8
    ) THEN  GO TO 1530
1690 IF ri=8 THEN  LET h$(a)=s$:
     GO SUB 3000
1700 IF ri=9 THEN  IF LEN s$>m(9
    ) THEN  GO TO 1530
1710 IF ri=9 THEN  LET i$(a)=s$:
     GO SUB 3000
1720 IF ri=10 THEN  IF LEN s$>m(
    10) THEN  GO TO 1530
1730 IF ri=10 THEN  LET l$(a)=s$
    : GO SUB 3000
1740 INPUT "Vuoi variare un altr
    o campo? ";r$
1750 IF r$<>"n" AND r$<>"s"
     THEN  GO TO 1740
1760 IF r$="s" THEN  GO TO 1520
1770 RETURN
2000 REM cancellazione record
2010 LET cancellazione=1:
     GO SUB 1000
2015 PAUSE 200
2020 FOR x=a TO (ctr-1)
2030 LET a$(x)=a$(x+1)
2040 LET b$(x)=b$(x+1)
2050 LET c$(x)=c$(x+1)
2060 LET d$(x)=d$(x+1)
2070 LET e$(x)=e$(x+1)
2080 LET f$(x)=f$(x+1)
2090 LET g$(x)=g$(x+1)
2100 LET h$(x)=h$(x+1)
2110 LET i$(x)=i$(x+1)
2120 LET l$(x)=l$(x+1)
2130 NEXT x
2135 LET ctr=ctr-1
2137 PRINT #0; FLASH 1;" CANCE
     LLAZIONE EFFETTUATA! ":
      PRINT #1;" premi un tasto
     per proseguire ": PAUSE 0
2140 RETURN
2500 REM salvataggio dati
2510 INPUT "vuoi cambiare il nom
     e di registrazione? (s/n) "
     ;r$
2515 IF r$<>"s" AND r$<>"n"
      THEN  GO TO 2510
2520 IF r$="n" THEN  SAVE z$
      LINE 190: RETURN
```

```
2530 INPUT "Nuovo nome? (10 posi
     zioni max.) ";z$
2540 IF LEN z$>10 THEN  GO TO 25
     30
2550 SAVE z$ LINE 190
2555 VERIFY z$
2560 RETURN
3000 REM video
3005 PAPER 1: CLS :
3010 PRINT TAB 14; FLASH 1;
      PAPER 7;"RECORD": PAPER 7:
      PRINT
3015 FOR x=1 TO ca
3020 PRINT AT x*2,0; PAPER 6;"
          (";x;")";n$(x);"
          "
3030 IF x=1 THEN  PRINT AT 3,0;a
     $(a)
3040 IF x=2 THEN  PRINT AT 5,0;b
     $(a)
3050 IF x=3 THEN  PRINT AT 7,0;c
     $(a)
3060 IF x=4 THEN  PRINT AT 9
     ,0;d$(a)
3070 IF x=5 THEN  PRINT AT 1
     1,0;e$(a)
3080 IF x=6 THEN  PRINT AT 1
     3,0;f$(a)
3090 IF x=7 THEN  PRINT AT 1
     5,0;g$(a)
3100 IF x=8 THEN  PRINT AT 1
     7,0;h$(a)
3110 IF x=9 THEN  PRINT AT 1
     9,0;i$(a)
3120 IF x=10 THEN  PRINT
     AT 21,0;l$(a)
3130 NEXT x
3140 RETURN
```



Slot Machine



di **Carlo** e **Mauro Corti**

Vivete in casa vostra le follie di Las Vegas, con gli amici e... lo Spectrum!

In un mondo afflitto da così tanti problemi c'è anche il momento per giocare e noi abbiamo concepito questo programma ponendoci il traguardo di farvi divertire il più possibile. Bando alle ciance e via con la descrizione!!
L'ambiente che ci ricorda questo programma è quello che ognuno di noi vorrebbe vedere e di cui vorrebbe far parte almeno una volta nella vita: "Las Vegas". Si tratta infatti di una coloratissima slot machine con la quale potremo iniziare il nostro viaggio fantastico nelle sale da gioco. Dal momento in cui si tira la leva cinque rulli ci mostrano a turno campane, mele succose, ciliegie, banane e fragole lasciandoci fino all'ultimo col fiato sospeso nella speranza di aver azzeccato una combinazione.
Quando i cinque fatidici rulli smettono di ruotare, subito appare sullo schermo la combinazione che abbiamo ottenuto e soprattutto se la nostra giocata ci ha fruttato qualcosa. Come accade però in qualsiasi gioco di fortuna non sempre si può vincere, ma non tutto è perduto poiché quasi come Superman nel momento del bisogno può intervenire l'opzione HOLD (tradotta in «blocca!») che ci permette di bloccare i rulli che potrebbero servire per un eventuale combinazione vincente. Migliore è la combinazione (minimo 2 campane accostate), maggiore naturalmente è l'ammontare della vincita. Volendo possono partecipare fino a quattro giocatori contemporaneamente, e vinca il migliore!

Descrizione delle variabili

P$	Prende come valore stringa il numero dei giocatori (max 4)
P	Trasforma la stringa PS in valore numerico
DIM M (P)	In questo modo si inizializza un vettore che ha come valore il numero dei giocatori e ne contiene il punteggio
E	Fa parte di un ciclo FOR-NEXT con limite P
X	Fa parte di un ciclo FOR-NEXT con limite P e si incrementa di una unità dopo ogni giocata
V	Serve per il controllo del numero dei giocatori rimasti senza soldi
A$	contiene i diversi caratteri grafici (DIM A$ (5,4))
B$	lo stesso di AS (DIM B$ (5,4))
A	serve per la stampa di rulli (DIM A (5))
Q	serve per tenere bloccati i rulli dell'HOLD (DIM Q (5))
W	cifra vinta
Q$	serve per il numero del rullo dell'HOLD
M$	contiene i nomi dei giocatori (DIM M$ (P,10))

Descrizione del programma

22-34	chiedono il numero e i nomi dei giocatori
48-60	gestiscono la casualità dell'opzione HOLD e il programma
100-200	opzione HOLD
240-280	stampa rulli
311-480	rotazione dei rulli
485-700	controllano le combinazioni ottenute, le vincite e il punteggio
1040-8400	stampano la slot machine, inizializzano tutte le variabili (comprese quelle di stampa) e controllano la risposta al messaggio "vuoi giocare ancora (s/n)?"

Come salvare il programma

Per prima cosa digitate il programma principale stando bene attenti a scrivere le lettere corrispondenti ai caratteri grafici in modo grafico.
La seconda operazione consiste nel salvarlo così:
SAVE "SLOT MAC." LINE 1
Il computer si metterà quindi in attesa di un altro programma. Digitate allora il programma di creazione di caratteri grafici, dando alla fine RUN 1160 e salvandolo di seguito al primo in questo modo:
SAVE "carslot" CODE USR "a", 168

Slot Machine versione per ZX Spectrum 16K/48K

Programma principale

```
1 LOAD ""CODE : CLS
2 PRINT AT 0,10; FLASH 1;
   INK 2;"SLOT MACHINE":
3 PRINT  INK 3; FLASH 0;"La s
  lot machine e' pronta per
   essere usata,ma voi forse
  no.  Vorrei sprecare due p
  arole      sull'opzione ""B
  LOCCA!"" che vi   aiutera'
  nei momenti di bisogno.Quan
```




```
  do essa entra in funzione
   voi potete bloccare da 1 a
   5    rulli contemporaneame
  nte "
 4 PRINT  INK 3;"semplicemente
   schiacciando i  tasti co
  rrispondenti da 1 a 5.  Ora
   siete pronti anche voi!"
 5 PRINT  INK 2;AT 14,7;"BUON
  DIVERTIMENTO!": PRINT
   INK 1;AT 20,9;"PREMI  UN TA
  STO"
 6 FOR A=0 TO 7
 7 BORDER A
 8 PAUSE 3
 9 IF INKEY$="" THEN  GO TO 13
10 IF INKEY$<>"" THEN
   GO TO 15
11 GO TO 6
13 NEXT A
14 GO TO 6
15 REM *****************
  ****
16 REM     by C.Corti
17 REM *****************
  ******
18 REM       2/4/84
19 REM *********************
  **
20 REM      SLOT MACHINE
  {4G8}
21 REM *********************
  **
22 CLS : BORDER 7: PRINT AT 10
  ,0; PAPER 4; INK 0;"In quan
  ti volete giocare(1-4)?"
23 LET P$=INKEY$
24 IF P$<CHR$ 49 OR P$>CHR$ 52
   THEN  GO TO 23
25 IF P$<>"" THEN  GO TO 27
26 GO TO 23
27 LET P=VAL P$
28 DIM M(P): DIM M$(P,10)
29 LET M=50
30 POKE 23658,8
31 FOR Z=1 TO P
32 INPUT "Scrivi il tuo nome(m
  ax 10 lett.)"; LINE M$(Z)
33 NEXT Z
34 POKE 23658,0
48 GO SUB 720:
   REM GRAFICA
49 GO SUB 290: REM ROTAZIONE

50 IF RND>.75  THEN  GO SUB 70
  : REM HOLD
51 IF M(X)=0 AND V<P THEN
   PRINT AT 0,0;"
                           "
55 PRINT AT X*2,17; INK 2;M$(X
  );AT x*2,27;"$";M(X)
60 NEXT X: GO TO 312
80 REM      HOLD

100 FOR T=1 TO 7: BEEP .1,50-
   T*7: NEXT T
110 PRINT AT 1,22; FLASH 1;
    INK (RND*4);"BLOCCA!"
115 LET Q$=INKEY$
121 IF Q$=CHR$ 13 THEN  GO TO 1
   90
124 IF Q$<CHR$ 49 OR Q$>CHR$ 53
    THEN  GO TO 115
125 IF Q$<>"" THEN  GO TO 130

126 GO TO 115
130 LET Q=VAL Q$
150 LET Q(Q)=Q
160 BEEP 0.08,4+Q*2
170 PRINT AT 11+Q,20; INK (RND*
   4); FLASH 1;"Bloccato> ";Q
180 GO TO 115
190 BEEP 0.08,15
191 PRINT AT 11,22;"         ";
   AT 12,20;"           ";
   AT 13,20;"           ";
   AT 14,20;"        ";AT 1
   5,20;"          ";AT 16,20
   ;"           "
195 PRINT AT 19,0;"
    "
200 GO TO 313
```

```
220 REM      STAMPA RUOTE
    {6G8}
240 PRINT AT 6,1;A$(A(1));AT 7,
    1;B$(A(1))
250 PRINT AT 6,4;A$(A(2));AT 7,
    4;B$(A(2))
260 PRINT AT 6,7;A$(A(3));AT 7,
    7;B$(A(3))
270 PRINT AT 6,10;A$(A(4));AT 7
    ,10;B$(A(4))
275 PRINT AT 6,13;A$(A(5));AT 7
    ,13;B$(A(5))
280 RETURN
300 REM       ROTAZIONE{8G8}
311 FOR X=1 TO P: LET M(X)=50:
    NEXT X
312 FOR X=1 TO P
314 IF V=P OR V>P THEN  PRINT
    AT 20,0;"Vuoi giocare ancor
    a( / )?"; INVERSE 1;
    FLASH 1;AT 20,20;"s";
    INVERSE 0;AT 20,22;"n":
    GO TO 8200
315 IF M(X)=0 THEN  GO TO 55
316 PRINT AT X*2,17; INK RND*4;
    FLASH 1;M$(X)
329 LET M(X)=M(X)-10
330 LET W=0
340 LET Q$="": LET P$=""
350 PRINT AT 20,0; FLASH 1;
    PAPER 1; INK 7;" Premi ENT
    ER per tirare la leva"
360 IF INKEY$<>"" THEN
    GO TO 360
370 IF INKEY$="" THEN  GO TO 37
    0
375 IF INKEY$>CHR$ 13 OR
    INKEY$<CHR$ 13 THEN
    GO TO 360
380 PRINT AT 20,0;"
                          "
390 PRINT AT X*2,28; FLASH 1;
    INVERSE 1; INK (RND*4);M(X
    ); FLASH 0; INVERSE 0;" "
400 FOR Z=1 TO 8
410 FOR B=1 TO 5
420 IF Q(B)=B THEN  GO TO 450
430 BEEP .008,Z*B
440 LET A(B)=INT (RND*5)+1
450 NEXT B
460 GO SUB 210
470 NEXT Z
480 DIM Q(5)
482 REM        PUNTEGGIO
485 IF A(1)=A(2) AND A(2)=A(3)
    AND A(3)=A(4) AND A(4)=A(5
    ) THEN  LET Q$=" JACKPOT!!
    ": LET W=200: GO TO 520
490 IF (A(1)=A(2) AND A(2)=A(3)
    AND A(3)=A(5)) OR (A(1)=A(
    2) AND A(2)=A(4) AND A(4)=A
    (5)) OR (A(1)=A(2) AND A(2)
    =A(3) AND A(3)=A(4)) OR (A(
    1)=A(3) AND A(3)=A(4) AND A
    (4)=A(5)) OR (A(2)=A(3)
    AND A(3)=A(4) AND A(4)=A(5
    )) THEN  LET Q$="4 DI UN TI
    PO!": LET W=100: GO TO 520
500 IF (A(1)=A(3) AND A(3)=A(4)
    ) OR (A(1)=A(2) AND A(2)=A(
    3)) OR (A(2)=A(3) AND A(3)=
    A(4)) OR (A(2)=A(3) AND A(3
    )=A(4)) OR (A(1)=A(2) AND A
    (2)=A(4)) THEN  LET Q$="3 D
    I UN TIPO!": LET W=35
505 IF (A(1)=A(4) AND A(4)=A(5)
    ) OR (A(2)=A(4) AND A(4)=A(
    5)) OR (A(3)=A(4) AND A(4)=
    A(5)) OR (A(1)=A(3) AND A(3
    )=A(5)) OR (A(2)=A(3) AND A
    (3)=A(5)) OR (A(1)=A(2)
    AND A(2)=A(5)) THEN  LET Q
    $="3 DI UN TIPO!": LET W=35
507 IF (A(1)=A(4) AND A(4)=A(5)
    ) OR (A(2)=A(4) AND A(4)=A(
    5)) OR (A(3)=A(4) AND A(4)=
    A(5)) OR (A(1)=A(3) AND A(3
    )=A(5)) OR (A(2)=A(3) AND A
    (3)=A(5)) OR (A(1)=A(2)
    AND A(2)=A(5)) THEN  LET Q
    $="3 DI UN TIPO!": LET W=35
510 IF A(1)=2 AND A(2)=2 OR A(2
    )=2 AND A(3)=2 OR A(3)=2
    AND A(4)=2 OR A(4)=2 AND A
    (5)=2 THEN  LET P$="BONUS!
    !": LET W=W+15
520 FOR G=1 TO 30
530 BEEP .008,50-G: BEEP .008,5
    0
540 NEXT G
550 LET M(X)=M(X)+W
560 IF Q$="" THEN  GO TO 630
570 PRINT AT 19,1; BRIGHT 1;
    FLASH 1; INK 2; PAPER 4;Q$
580 PRINT TAB 1; BRIGHT 1;
    FLASH 1; INVERSE 1; INK 2;
    PAPER 4;Q$
```

Slot Machine

```
590 IF W>0 THEN  PRINT AT 10,3;
     BRIGHT 1; INK 2;"VINCI $";
    W
600 IF P$="" THEN  GO TO 630
610 PRINT AT 10,20; FLASH 1;
     INK 5; PAPER 3;P$
620 PRINT AT 11,20; INVERSE 1;
     FLASH 1; INK 5; PAPER 3;P$
630 FOR T=1 TO 100: BEEP .008,T
    -40: NEXT T
640 PRINT AT X*2,28; FLASH 1;
     INVERSE 1; INK RND*4;M(X);
     INVERSE 0
650 PRINT AT 19,1;"
     ";TAB 1;"
    "
660 PRINT AT 10,2; INK 0;"
    {13SG8}"
670 PRINT AT 10,20;"     ";
    AT 11,20;"     "
680 IF M(X)<1 THEN  LET V=V+1:
     PRINT AT 0,0; FLASH 1;M$(X
    );"NON PUOI PIU' GIOCARE":
     FOR z=4 TO 1 STEP -1:
     BEEP 1,z: NEXT z: GO TO 51

690 IF M(X)>499 THEN  PRINT
    AT 0,0; FLASH 1;M$(X);"MI H
    AI SBANCATO!!": PRINT AT 20
    ,0;"Vuoi giocare ancora( /
    )?";AT 20,20; INVERSE 1;
     FLASH 1;"s"; INVERSE 0;
    AT 20,22;"n": GO TO 8200
700 GO TO 50
730 REM     VARIABILI/GRAFICA
    {3G8}
1040 PAPER 7: CLS : BORDER 7
1050 PRINT AT 1,1; INK 0;"
    {11SG8}"
1060 PRINT TAB 0; INK 0;"{2SG8}"
    ; INK 4; PAPER 2; FLASH 1;"
    {9SG8}"; FLASH 0; PAPER 7;
     INK 0;"{2SG8} {SG8}"
1070 PRINT TAB 0; INK 0;"
    {2SG8}"; INVERSE 1; INK 4;
     PAPER 2; FLASH 1;"{9SG8}";
     INVERSE 0; FLASH 0;
     PAPER 7; INK 0;"{2SG8}
    {SG8}"
1080 FOR Z=1 TO 11
1090 PRINT TAB 0;"{16SG8}"
1100 NEXT Z
1110 PRINT TAB 0;"{SG8}";
      PAPER 6; INK 3; FLASH 1;"
     {11SG8}"; FLASH 0; PAPER 7;
      INK 0;"{4SG8}"
1120 PRINT TAB 0;"{SG8}";
      PAPER 6; INK 3; FLASH 1;
      INVERSE 1;"{11SG8}";
      FLASH 0; INVERSE 0;
      PAPER 7; INK 0;"{4SG8}"
1130 PRINT TAB 0;"{16SG8}"
1140 FOR E=1 TO P
1150 PRINT AT E*2,17; INK 2;M$(E
     );AT E*2,27;"$";M
1160 NEXT E
2000 PRINT #0; INK 2; INVERSE 1;
     "1984 © SLOT MACHINE © by C
     .Corti"; INVERSE 0
4000 LET X=0
4005 LET V=0
4010 DIM A$(5,4): DIM B$(5,4):
      DIM A(5): DIM Q(5)
5050 REM
5060 REM
5070 REM
5080 REM "A" e "B" in modo G
5085 REM
5090 LET A$(1)=CHR$ 16+CHR$ 2+"A
     "+"B"
5200 REM
5500 REM "C" e "D" in modo G

5510 REM
6000 LET B$(1)=CHR$ 16+CHR$ 2+"C
     "+"D"
6010 REM
6030 REM "E" e "F" in modo G
6050 REM
6080 LET A$(2)=CHR$ 16+CHR$ 5+"E
     "+"F"
6081 REM
6083 REM "G" e "H" in modo G

6087 REM
6090 LET B$(2)=CHR$ 16+CHR$ 5+"G
     "+"H"
6095 REM
7000 REM "J" e "K" in modo G

7050 REM
7070 LET A$(3)=CHR$ 16+CHR$ 3+"J
     "+"K"
7071 REM
7073 REM "L" e "M" in modo G
```

```
7077 REM
7080 LET B$(3)=CHR$ 16+CHR$ 3+"L
     "+"M"
7090 REM
8000 REM "N" e "O" in modo G

8040 REM
8060 LET A$(4)=CHR$ 16+CHR$ 6+"N
     "+"O"
8061 REM
8063 REM "P" e "Q" in modo G

8067 REM
8070 LET B$(4)=CHR$ 16+CHR$ 6+"P
     "+"Q"
8080 REM
8090 REM "R" e "S" in modo G

8100 REM
8150 LET A$(5)=CHR$ 16+CHR$ 2+"R
     "+"S"
8151 REM
8153 REM "T" e "U" in modo G

8157 REM
8160 LET B$(5)=CHR$ 16+CHR$ 2+"T
     "+"U"
8165 REM
8170 RETURN
8200 IF INKEY$="S" OR INKEY$="s"
      THEN  GO TO 15
8300 IF INKEY$="N" OR INKEY$="n"
      THEN  STOP
8400 GO TO 8200
```

Slot Machine
versione per ZX Spectrum 16K / 48K

Caratteri grafici

```
   1 REM         PARTE UDG

   2 REM
   3 REM
1150 REM MELA
1160 DATA 0,0,121,254,0,16,63,25
     4,0,16,63,254,15,32,31,254
1170 DATA 31,240,5,252,51,252,7
     ,248,52,254,1,224,101,254,0
     ,0
1180 REM CAMPANA
1190 DATA 0,128,7,240,0,128,15,2
     48,1,192,15,248,3,224,31,25
     2
1200 DATA 3,224,0,128,3,224,1,19
     2,3,224,1,192,3,224,0,0
1210 REM CILIEGIA
1220 DATA 0,0,1,136,0,2,26,8,56,
     12,60,16,124,52,126,208
1230 DATA 127,204,61,224,124,20,
     25,240,124,36,1,224,56,72,0
     ,224
1232 REM BANANA
1235 DATA 0,0,0,112,4,0,0,112,2,
     0,0,112,1,128,0,224
1236 DATA 1,192,1,192,0,224,3,12
     8,0,112,6,0,0,112,0,0
1237 REM FRAGOLA
1238 DATA 0,0,63,188,7,32,26,236
     ,9,80,15,248,1,128,5,176
1239 DATA 7,224,7,240,29,248,3,2
     24,55,188,1,192,61,252,0,0
5000 RESTORE
5010 FOR Z=0 TO 7
5020 READ A: READ B: READ C:
      READ D
5030 POKE USR "A"+Z,A
5050 POKE USR "B"+Z,B
5060 POKE USR "C"+Z,C
5070 POKE USR "D"+Z,D
5080 NEXT Z
6010 FOR Z=0 TO 7
6020 READ E: READ F: READ G:
      READ H
6030 POKE USR "E"+Z,E
6040 POKE USR "F"+Z,F
6050 POKE USR "G"+Z,G
6060 POKE USR "H"+Z,H
6070 NEXT Z
7000 FOR Z=0 TO 7
7010 READ J: READ K: READ L:
      READ M
7020 POKE USR "J"+Z,J
7030 POKE USR "K"+Z,K
7040 POKE USR "L"+Z,L
7050 POKE USR "M"+Z,M
7060 NEXT Z
7090 FOR Z=0 TO 7
8000 READ N: READ O: READ P:
      READ Q
8010 POKE USR "N"+Z,N
8020 POKE USR "O"+Z,O
8030 POKE USR "P"+Z,P
8040 POKE USR "Q"+Z,Q
8050 NEXT Z
8080 FOR Z=0 TO 7
8090 READ R: READ S: READ T:
      READ U
8100 POKE USR "R"+Z,R
8110 POKE USR "S"+Z,S
8120 POKE USR "T"+Z,T
8130 POKE USR "U"+Z,U
8140 NEXT Z
```

Soluzione dei triangoli

di **Francesco Santi**

"Dati tot elementi di un triangolo calcolarne il perimetro e l'area..."

Questo programma permette di determinare il perimetro, l'area, i lati incogniti di un triangolo qualsiasi di cui si conoscano almeno tre elementi.
I casi che si possono presentare sono i seguenti:
I) tre lati noti
II) due lati e l'angolo compreso noti
III) due lati e due angoli noti
IV) un lato e due angoli noti.
È un programma particolarmente utile per tutti gli studenti medi (non ditelo al professore!). Comunque può trovare parecchie altre applicazioni: infatti, una qualsiasi figura piana non curva può essere ridotta ad un insieme di triangoli e, ad esempio, questo è nato per determinare le aree di stanze dalla pianta irregolare.
Il programma inizia, dopo la presentazione, a richiedere i dati del triangolo, per forza di cose è necessario inserire i lati incogniti e gli angoli incogniti ponendoli uguali a zero.
Il secondo passo del programma, una volta acquisti i dati, consiste nel controllo dei dati stessi. Questo al fine di determinare la loro eventuale insufficienza od erroneità, per poter poi determinare la strategia di soluzione. Attenzione, però: introducendo dati inverosimili è possibile, a volte, il blocco del programma. In questo caso ridate pure tranquillamente RUN e ricominciate da capo.
Le due routine RUOTA LATI e RUOTA ANGOLI servono appunto a ruotare i lati e gli angoli allo scopo di avere sempre come primo lato quello noto.
Ultima cosa: gli angoli vengono inseriti in gradi sessagesimali.



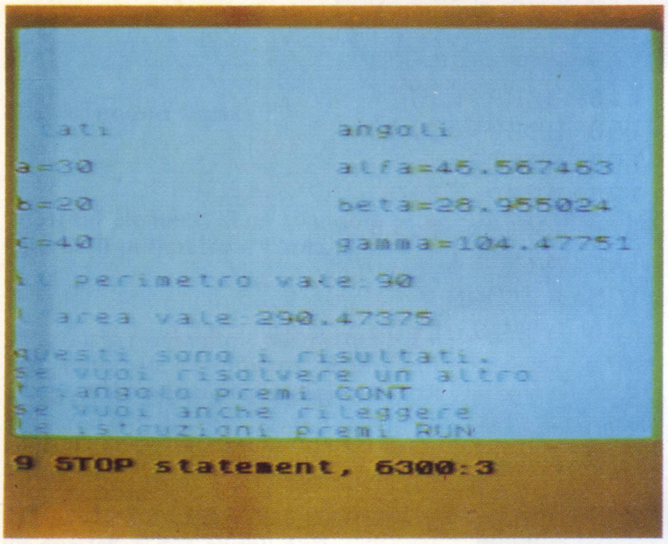
Una volta scelta la strategia il triangolo viene risolto facendo uso alle funzioni trigonometriche dello SPECTRUM.
Quindi, per poter disegnare il triangolo, il programma procede a un test nel quale definisce il parametro K col quale verranno poi moltiplicati i risultati; successivamente viene disegnato sullo schermo un triangolo di lati = K*lati reali.
Infine sullo schermo appaiono i risultati finali.
Il programma abbanda di REM esplicative, e la sua struttura è notevolmente lineare così che la sua comprensione non determinerà grandi difficoltà. Potrà anche esservi d'aiuto per rispolverare le vostre conoscenze di trigonometria...

Soluzione dei triangoli versione per ZX Spectrum 16K/48K

```
10 BORDER 6: INK 2: PAPER 5:
   FLASH 1: CLS
20 REM SOLUZIONE DEI TRIANGOLI
30 PRINT AT 5,9;"{SG2}{2SG8}
   {SG7} {SG2}{2SG8}{SG7}
   {SG2}{2SG8}{SG7}"
40 PRINT AT 6,9;"{SG8}{G2}
   {SG8}{SG3}  {SG8}{G2}{G7}
   {SG5}"
```

```
51 PRINT AT 7,9;"{SG8}{SG7}
   {G4}{SG7} {SG8}{G3}
   {SG8}{SG7}{SG2}{SG5}"
52 PRINT AT 8,9;"{G7}{2SG8}
   {G2} {G7}{2SG8}{G2} {G7}
   {2SG8}{G2}"
53 PRINT AT 13,3;"{SG2}{SG7}
   {G4}{SG1} {SG2}{2SG8}{SG7}
   {G4}{2SG8}{SG7} {SG2}{SG8}
   {SG1}  {SG8}  {SG2}{SG1}"
54 PRINT AT 14,3;"{4SG8}
   {SG8}{SG3}   {G5}{SG5}
   {SG8}{SG3}{SG8}  {SG8}
   {G5}{SG4}{G7}{SG5}"
55 PRINT AT 15,3;"{SG8}{G1}
   {G2}{SG8} {SG8}{G3}  {G5}
   {SG5}  {SG8}{G7}{SG1}
   {SG8}  {SG8}{SG1}{SG2}
   {SG8}"
56 PRINT AT 16,3;"{SG8}
   {SG8} {G7}{2SG8}{G2} {G5}
   {SG5}  {SG8} {G7}{SG5}
   {SG8}  {SG8}{G2}{G1}{SG8}"
57 LET T=2: GO SUB 100
60 FLASH 0: BRIGHT 1: CLS
65 PRINT AT 2,7;"* PRESENTAZIO
   NE *"
66 PRINT AT 5,2;"questo progra
   mma ti permette"'"di risolv
   ere un qualsiasi"'"triangol
   o conoscendone almeno 3"'"e
   lementi"
67 PRINT AT 11,2;"presta molta
    attenzione alle"'"istruzio
   ni che seguono"''"" premi
    Y per continuare"
68 PRINT AT 21,1;" "
69 PRINT
70 PRINT "  RICORDATI:"
71 PRINT "  il primo lato si c
   hiama  a"'"  il secondo
             b"'"  il terzo
                c"
72 PRINT
73 PRINT "  il primo angolo e'
    quello"'" compreso fra i l
   ati "'" b e c"
74 PRINT
75 PRINT " il secondo angolo
    e' quello"'" copreso fra i
    lati"'" a e c"
76 PRINT
77 PRINT "  il terzo angolo e'
    quello"'" compreso fra i l
   ati"'" a e b"
78 PRINT
79 PRINT "  premi per continua
   re"
80 GO SUB 100
82 PAUSE 0
85 CLS
86 GO TO 800
100 REM routine suono
110 FOR i=1 TO T
120 BEEP .1,-7
130 BEEP .1,-4
140 BEEP .1,2
150 BEEP .1,12
160 NEXT i
170 BEEP .1,2
180 BEEP .1,-4
190 NEXT i
195 RETURN
800 REM lettura dei dati
805 LET P=180/PI
810 INPUT "Lunghezza lato a",A
820 INPUT "Lunghezza lato b",B
830 INPUT "Lunghezza lato c",C
840 LET a$="ampiezza dell'angol
    o"
850 LET b$=",angolo compreso fr
    a i lati"
860 PRINT "Quant'e' l'";a$';"AL
    FA";b$';"B e C"
870 INPUT ALFA: CLS
875 LET ALFA=ALFA/P
880 PRINT "Quant'e' l'";a$';"BE
    TA";b$';"A e C"
890 INPUT BETA: CLS
895 LET BETA=BETA/P
900 PRINT "Quant'e' l'";a$';"GA
    MMA";b$';"A e B"
905 INPUT GAMMA: CLS
906 LET GAMMA=GAMMA/P
910 REM scelta della soluzione
915 GO SUB 5350
920 IF A<>0 AND B<>0 AND C
    <>0 THEN  GO TO 1000
930 IF A<>0 AND B<>0 AND GA
    MMA<>0 THEN  GO TO 2000
940 IF A<>0 AND B<>0 AND AL
    FA<>0 AND BETA<>0
     THEN  GO TO 3000
955 IF A<>0 AND BETA<>0
     AND GAMMA<>0 THEN
     GO TO 4000
960 IF A<>0 AND ALFA<>0
     AND BETA<>0 THEN
     GO TO 4000
965 IF A<>0 AND ALFA<>0
     AND GAMMA<>0 THEN
     GO TO 4000
```

Soluzione dei triangoli




```
 970 CLS : PRINT AT 11,8;
     INVERSE 1;"dati insufficie
     nti!"'"ricominciamo a legge
     re i dati": LET T=10:
     GO SUB 100
 971 PAUSE 0: CLS : INVERSE 0
 975 GO TO 800
1000 REM caso dei tre lati noti
1010 LET ALFA=ACS ((B↑2+C↑2-A↑2)
     /(2*B*C))
1020 LET BETA=ACS ((A↑2+C↑2-B↑2)
     /(2*A*C))
1030 LET GAMMA=ACS ((A↑2+B↑2-C↑2
     )/(2*A*B))
1040 GO TO 6000
2000 REM caso due lati e angolo
     compreso noti
2010 LET C=SQR ((A↑2+B↑2-2*A*B*
     COS (GAMMA)))
2020 GO TO 1010
3000 REM caso due lati e due ang
     oli noti
3010 LET GAMMA=PI-ALFA-BETA
3020 GO TO 2000
4000 REM caso un lato e due ango
     li noti
4010 IF ALFA=0 THEN  LET ALFA=
     PI-BETA-GAMMA
4020 IF BETA=0 THEN  LET BETA=
     PI-ALFA-GAMMA
4030 IF GAMMA=0 THEN  LET GAMMA=
     PI-ALFA-BETA
4040 LET A2=A/SIN (ALFA)
4050 LET B=A2*SIN (BETA)
4060 LET C=A2*SIN (GAMMA)
4070 GO TO 6000
5000 CLS : PRINT AT 10,10;"DATI"

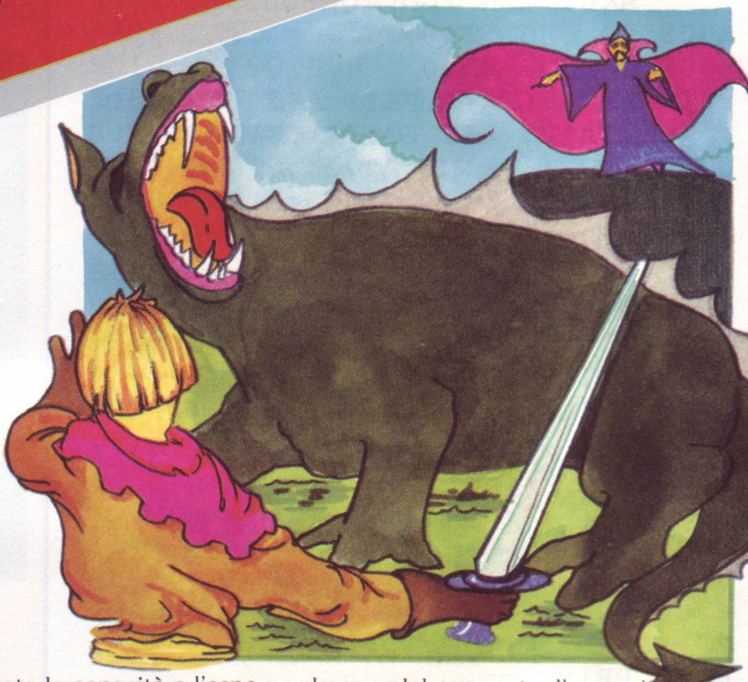
5001 PRINT AT 11,10;"ERRATI"
5002 LET T=50: GO SUB 100
5010 GO TO 800
5050 LET B=A: LET A=C: LET C=0
5055 GO SUB 5390
5060 GO TO 920
5100 LET A=C: LET C=B: LET C=0
5110 GO SUB 5390
5115 GO TO 920
5300 REM coordinate dei vertici
5305 LET XB=50: LET YB=50
5310 FOR i=21 TO -21 STEP -1
5315 LET K=i
5316 IF 50+K*A<255 AND 50+K*A-K*
     B*COS (GAMMA)<255 AND 50+K*
     B*SIN (GAMMA)<175 AND 50+K*
     C*COS (BETA)>0 THEN
      GO TO 5325
5320 NEXT i
5325 LET XC=K*A: LET YC=0
5330 LET XA=-K*B*COS (GAMMA):
      LET YA=K*B*SIN (GAMMA)
5335 LET XXB=-K*C*COS (BETA):
      LET YYB=-K*C*SIN (BETA)
5340 RETURN
5350 REM routine ruota lati
5355 IF A<>0 AND C=0 THEN
      RETURN
5360 IF A<>0 AND B<>0 AND C
     <>0 THEN  RETURN
5365 IF A<>0 AND B=0 THEN
      GO TO 5050
5370 IF A=0 AND B=0 AND C=0
      THEN  GO TO 970
5375 IF A=0 THEN  GO TO 5100
5380 RETURN
5390 REM routine ruota angoli
5391 LET DELTA=ALFA
5392 LET ALFA=BETA: LET BETA=GAM
     MA
5393 LET GAMMA=DELTA
5394 RETURN
5500 REM routine disegna triango
     lo
5505 PLOT XB,YB
5510 DRAW XC,YC: DRAW XA,YA:
      DRAW XXB,YYB
5511 PAUSE 200: CLS
5512 RETURN
6000 CLS : REM conclusione
```

```
6005 IF A>B+C OR B>A+C OR C>A+B
     THEN  GO TO 5000
6006 IF ALFA>PI OR BETA>PI
     OR GAMMA>PI THEN  GO TO 5
     000
6007 IF ALFA<0 OR BETA<0 OR GAMM
     A<0 THEN  GO TO 5000
6008 GO SUB 5300
6009 GO SUB 5500
6010 LET ALFA=ALFA*P: LET BETA=B
     ETA*P: LET GAMMA=GAMMA*P
6200 CLS : PRINT AT 5,1;"lati","
     angoli"
6205 PRINT
6210 PRINT "a=";A,"alfa=";ALFA
6220 PRINT
6230 PRINT "b=";B,"beta=";BETA
6240 PRINT
6250 PRINT "c=";C,"gamma=";GAMMA

6260 PRINT
6261 PRINT "il perimetro vale:";
     A+B+C
6262 PRINT
6265 PRINT "l'area vale:";A/2*B*
     SIN (GAMMA/P)
6266 PRINT
6270 PRINT "questi sono i risult
     ati.""""se vuoi risolvere un
      altro""triangolo premi CO
     NT"
6290 PRINT "se vuoi anche rilegg
     ere""""le istruzioni premi R
     UN"
6300 LET T=10: GO SUB 100:
     STOP
6305 CLS
6310 GO TO 800
```

Warlock

di **Hugh Walker**
trad. e adatt. di
Paolo Maffei



Pericolo, magia, mostri e tesori vi aspettano in questo super gioco di avventura per lo Spectrum 48K

Uno dei problemi dei giochi di avventura consiste nel fatto che questi vengono spesso completati grazie alla conoscenza del programma introdotto. In questo programma sono stati adottati due metodi per eliminare, per quanto possibile, questo problema.
Prima di tutto sono stati eliminate i e REM illustrative e le linee sono state introdotte in maniera illogica; inoltre è stato fatto un grande uso di funzioni RND per mantenere viva la suspence durante il gioco.
Interpretate il ruolo di un giovane eroe il cui compito è quello di introdursi nel castello di Warlock e di trovare tutte le monete d'oro, usando le chiavi per aprire le porte chiuse e combattendo con i mostri di guardia al tesoro, armato solo di una spada, di uno scudo e del vostro... coraggio. Ogni chiave che trovate può esser usata una sola volta, ma non tutte le porte sono chiuse.
Per trovare le monete d'oro e le chiavi dovete aprire porte, esplorare corridoi e combattere, se necessario, con i mostri; potrebbe capitare comunque di non trovare nulla e ci verrebbe quindi offerta la possibilità di riposarvi per rimettervi in forma.
La vostra forza, infatti, è limitata e si riduce ad ogni combattimento.
All'inizio del gioco Warlock non è nel castello, ma, state attenti, potrebbe ritornare in qualsiasi momento e allora saranno guai... infatti , se Warlock vi sorprenderà, sarà molto difficile scappare e, inoltre, il malvagio potrebbe usare le sue arti magiche per trasformarvi in un piccolo animaletto, portandovi via tutti i punti-coraggio conquistati.
Quando avrete recuperato tutte le monete d'oro dovrete uscire dal castello. Per uscire dovrete attraversare una porta posta al livello (piano) 0.
Ogni volta che vi troverete di fronte ad un mostro o a Warlock stesso avrete due possibilità: "combattimento" o "fuga". Non avete la capacità e l'esperienza per uccidere gli avversari (ricordatevi che siete solo un giovane inesperto), ma potete solo costringerli alla fuga e ferirli.
Comunque non è sempre facile vincere i duelli: ricordatevi che ad ogni combattimento diminuisce la vostra forza e che quando questa si esaurisce completamente avete perso la partita.
Se decidete comunque di scappare verrete considerati dei codardi, perderete punti-coraggio, e rischierete di perdere qualche moneta d'oro e di essere intrappolati dal mostro.
All'interno del castello troverete le altre seguenti situazioni: "porta", "corridoio", "scala".
Solo il computer conosce l'esatto numero di monete d'oro da trovare di cui voi non sarete a conoscenza finché non le avrete recuperate tutte (sarà comunque un numero di monete compreso fra 1500 e 2500). Per registrare il programma, digitate GO TO 9000.
Buona fortuna!

Opzioni del programma

1 Porta	(potrebbe essere chiusa) avete due scelte: "apri" o "prosegui". Se aprite la porta potete trovare delle monete d'oro... o un mostro... o nulla. In questo caso dovete proseguire. Se la porta è chiusa e non ne avete la chiave dovete proseguire... con la possibilità di incontrare un mostro.
2-3	Scelte: "ti muovi", "esplori", "situazione". "ti muovi": muovendovi indietro o lateralmente può capitarvi di urtare il capo contro una trave causandovi una perdita di energia; muovendovi in avanti non correte questo pericolo perché riuscite a vedere la trave! "esplori": potete trovare delle monete d'oro, una chiave, del vino e del cibo per rimettervi in forma, un mostro, un'altra situazione o nulla, dovendo in tal caso così proseguire con i soliti rischi. "situazione": vi viene presentata una situazione in cui vi trovate in quel preciso istante: livello, mostri, etc.
3 Scala	3 scelte: "su", "giù", "prosegui". Non saprete se la scala va su o giù finché non avrete tentato. Se sbagliate dovete proseguire.

Descrizione delle variabili

matrici dimensionate

m$(9,15)	il nome di ogni mostro (e di Warlock)
a$(3,12)	le tre situazioni trovate (porta, corridoio (porta, corridoio, scala)
k$(3,6)	le tre piccole creature in cui potete essere trasformati.
m(9)	la resistenza di ogni mostro ai vostri colpi.
n(9)	la resistenza di ogni mostro a opporsi ai successivi colpi
g(9)	il valore che incrementa i punti-coraggio quando battete un mostro (linee 3090 e 3110).
z(11)	la lunghezza dei BEEP nella linea 4000.
y(11)	il tono dei BEEP nella linea 4000.

variabili semplici:

hi	record, aggiornato ad ogni partita.
hr (*)	punti-coraggio
hq	quoziente-coraggio, aumenta se si vince un duello.
ded	il numero dei mostri battuti.
ht	la potenza del vostro colpo durante il combattimento (scelta casualmente).
blo	il valore del decremento di resistenza del mostro ai colpi successivi.
mn	valore casuale tra 1 e 9 per scegliere il mostro da m$( ) e i relativi valori di m, n, g,.
str	la vostra forza, inizialmente al valore 500; questa diminuisce combattendo o ferendosi il capo; aumenta riposandosi.
wl	il valore di decremento della variabile 'str' quando vi ferite il capo contro una trave (scelto causalmente).
tres	il numero di monete d'oro da trovare.
tr	variabile logica che cambia corso al programma quando è stato raggiunto 'tres'; essa evita dei trovare altre monete e vi permette di uscire (linea 1630) quando siete al livello 0.
csh	il numero di monete in vostro possesso.
cn	il numero di monete di volta in volta trovate.
ky	il numero di chiavi in vostro possesso.
wy	valore casuale scelto tra 1 e 3 per scegliere a$ e per determinare il GOSUB nella linea 140
fl	il livello (piano) in cui vi trovate.
bp	variabile usata nelle linee 400-460; bp = + 1: su, bp = 1: giù; questa variabile inoltre modifica 'fl' e il tono dei Beep nella linea 460.

(*) I punti-coraggio sono così calcolati: csh + (ky + 5) + hq e vengono aggiornati ogni volta che si cambia situazione.
Nel corso del programma 'f', 'i' e 'h' vengono usate come variabili in diversi cicli FOR-NEXT.

Descrizione del programma

Linee	**Commento**
20	inizializza il record per le successive partite.
50-60	inizializzano ogni partita (routine completata con la chiamata della subroutine 5000-5500).
100-150	routine principale alla quale tutte le altre ritornano. Sceglie e scrive le diverse situazioni e determina il relativo GOSUB.

Situazioni: (linee 200-460)

200-220	scrive 'corridoio' e determina le scelte con GOTO 500, 600 e 700
300-330	scrive 'porta' e determina le scelte con GOTO 800 o 8200
400-460	scrive 'scala' e determina le scelte; routine completa.



Risposte alla routine 'corridoio' (linee 500-780)

500-520	opzione 'ti muovi' con INKEY$ "1" nella linea 220; presenta le diverse opzioni.
600	opzione 'esplori' con INKEY$ "2" nella linea 220; usa un GOT0 1650.
700-780	opzione 'situazione' con INKEY$ "3" nella linea 220; scrive ciò che si possiede, il punteggio, il livello in cui ci si trova e ritorna alla linea 200 quando si è pronti.

Risposte alla routine 'porta'

800	opzione 'apri' con INKEY$ "1" nella linea 320. Scelta casuale fra GOTO 1500 (porta chiusa) o GOTO 1600 (porta aperta).

Risposte alla routine 'esplori' e 'porta aperta'

100-1460	
100-1800	routine 'incontri un mostro'; sceglie il mostro, presenta
1650-8200	l'opzione 'combatti' (GOTO 3000) o 'scappi' (GOTO 8500), dà la possibilità al mostro di rubarvi alcune monete d'oro e, se incontrate Warlock, di essere trasformato in un animaletto (GOTO 1700).
1100-1150	routine 'trovi dell'oro', dalla linea 1650 o 3130. Controlla quante monete sono state trovate e le somma a quelle già in vostro possesso e scrive il totale; controlla inoltre se è stato raggiunto il numero di monete da trovare.
1200-1220	routine 'hai trovato una chiave'; dalla linea 1650 o 3130 Determina a caso se farvi trovare o meno una chiave; se la chiave viene trovata la somma alle precedenti.
1300	routine 'non c'è nulla'; utilizza le linee 8000-8210 per completare la routine.
1400-1460	routine 'riposati'; determina a caso se farvi riposare o meno; in caso affermativo aumenta la vostra forza.

Continuazione alla routine 'porta' (linee 1500-1650);

800	dalla routine 'porta chiusa'; controlla se avete la chiave; se non la possedete GOTO 6500. Se invece l'avete chiede se la volete usare; dopo di che decrementa il numero di chiavi in vostro possesso e va alla linea 1600.
1600-1650	routine 'porta aperta' (dalla linea 1550 o 800). Usa la linea 4100 per i Beep, controlla se la partita è stata terminata e sceglie a caso fra GOTO 1000, 1100, 1200, 1400.

Routine 'fine' (linee 1700-2050)

1700-1770	routine 'triste conclusione' (Warlock usa i suoi poteri magici). Determina e scrive k$ e usa le linee 1900-1950 per completare la routine.
1800-1880	routine 'finale vittorioso': scrive il punteggio ottenuto e usa le linee 1900-1950 per completare la routine.
1900-1950	routine 'record' e 'giochi ancora?'. Aggiorna il record e chiede se volete fare un'altra partita; se 'SI' va alla linea 50 altrimenti STOP.
2000-2050	routine 'non hai più forza'; Scrive il messaggio e usa le linee 1850-1950 per completare la routine.
2500-2570	routine 'hai trovato tutte le monete'. Scrive il messaggio e pone la variabile logica "tr" uguale a 1, quindi ritorna alla routine principale (linea 100).

Routine 'combattimento' (linee 3000-3130)

3000-3040	calcolano l'intensità del vostro colpo, decrementano la forza, controllano se hanno ferito il mostro; se si, GOTO 3050; se no GOTO 1050 per un altro tentativo.
3050-3130	calcolano in che misura avete ferito il mostro, decrementano la sua resistenza; se il mostro è ancora in grado di combattere si ritorna alla linea 1050, altrimenti

	aumentano i punti-coraggio; quindi GOTO 1100- o 1200.
3500	subroutine 'titolo'. Scrive il titolo usato in varie parti del programma.
4000-4500	routine Beep
5000-5190	routine d'inizio; Dimensionamento matrici, istruzioni READ per la lettura dei DATA nelle linee 7500-7800, istruzioni, inizializzazioni delle variabili; viene poi utilizzata la linea 5500 per continuare.
5500	usata per proseguire il programma
6000-6050	'il mostro ti porta via alcune monete d'oro' dalla linea 1040. Scrive il messaggio ed aggiorna la variabile "csh"; se erano state trovate tutte le monete viene resettata la variabile "tr".
6500	routine "no"; dalle linee 1510 e 1540. Utilizza GOTO 8000 per continuare.
8000	routine 'prosegui'; dalle linee 320, 430, 1300, 1460 e 6500; porta alla linea 8200.
8200-8210	viene determinato a caso se incontrare un altro mostro; dalle linee 330, 420, e 8000.
8500-8530	routine "scappi"; dalla linea 1070. Scrive "Codardo", decrementa i punti-coraggio, quindi sceglie fra:
8520	(a) vi vengono portate via alcune monete d'oro utiizzando GOTO 6010 per aggiornare la variabile "csh".
8540	(b) rimanete intrappolati; ritorna alla linea 1050 per un altro tentativo.
9000-9050	routine per il salvataggio su nastro del programma; questa routine è accessibile con BREAK: GOTO 9000.

Se avete uno Spectrum...

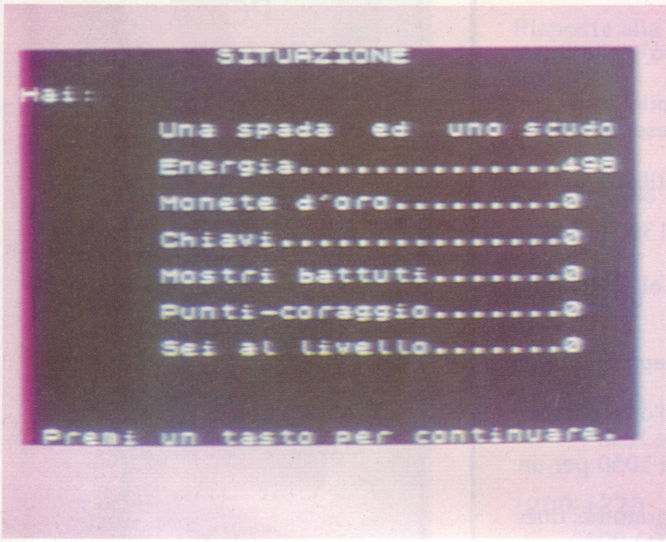
POKE 23561,0...35 regola il tempo di innesco dell'autorepeat della tastiera, e POKE 23562,0...5 si occupa della velocità dell'autorepeat stesso.

Warlock
versione per ZX Spectrum 48K

```
 10 REM ***** WARLOCK *****
 20 LET hi=0
 50 PAPER 0: INK 9: BORDER 0:
    CLS
 60 GO SUB 5000
100 LET wy=INT (RND*3)+1
110 CLS : PRINT "Hai trovato ";
    a$(wy)
120 LET hr=csh+ky*5+hq
140 GO SUB 100+100*wy
150 GO TO 100
200 PRINT '"(1)Ti muovi;(2)Espl
    ori;"'"(3)Situazione"
210 PAUSE 0: IF INKEY$<"1" OR
    INKEY$>"3" THEN  GO TO 210
220 GO TO 400+100*VAL INKEY$
300 PRINT '"(1) apri (2) prose
    gui"
310 PAUSE 0: IF INKEY$<"1" OR
    INKEY$>"2" THEN  GO TO 310
320 IF INKEY$="1" THEN  GO TO 8
    00
330 GO TO 8200
400 PRINT '"(1) su; (2) giu'; (
    3) prosegui"
410 PAUSE 0: IF INKEY$<"1" OR
    INKEY$>"3" THEN  GO TO 410
420 IF INKEY$="3" THEN  GO TO 8
    200
430 IF INT (RND*3) THEN
     PRINT '"Non puoi andare ";
    ("su'" AND INKEY$="1");("gi
    u'" AND INKEY$="2");:
     PAUSE 60: GO TO 8000
440 LET bp=(INKEY$="1")-(
    INKEY$="2")
450 LET fl=fl+bp: PRINT '" O.K.
     Ora sei al livello ";fl
460 FOR f=2 TO 30 STEP 2:
     BORDER INT (f/4): BEEP .05
    ,f*bp: NEXT f: BORDER 0:
     RETURN
500 PRINT '"(1)Avanti;   (2)Ind
    ietro"'"(3)Sinistra; (4)Des
    tra"
510 PAUSE 0: IF INKEY$>"1"
     AND INT (RND*2) THEN
     PRINT '"Hai urtato una tra
    ve"'"e ti sei ferito il ca
    po.": GO SUB 4500: LET wl=
    INT (RND*5)+1: LET str=str-
    wl*10: PRINT ".....Ahi!"
520 PAUSE 50: IF str<1 THEN
     GO TO 2000: RETURN
600 GO TO 1650
```

LOAD

Warlock




```
700 CLS : PRINT TAB 10;"SITUAZIONE"
710 PRINT '"Hai:"
720 PRINT 'TAB 7;"Una spada  ed uno scudo"
730 PRINT 'TAB 7;"Energia..............";str
740 PRINT 'TAB 7;"Monete d'oro.........";csh
750 PRINT 'TAB 7;"Chiavi..............";ky
760 PRINT 'TAB 7;"Mostri battuti.......";ded
770 PRINT 'TAB 7;"Punti-coraggio......";hr
780 PRINT 'TAB 7;"Sei al livello......";fl: GO SUB 5500: GO TO 200
800 GO TO 1500+100*INT (RND*2)
1000 LET mn=INT (RND*9)+1
1010 CLS : PRINT '" IMPROVVISAMENTE!"''" Incontri ";m$(mn)
1020 FOR f=2 TO 21: BEEP .04,10: BORDER INT (RND*7): BEEP .05,-10: NEXT f: BORDER 0
1030 IF NOT INT (RND*5) AND mn=9 THEN GO TO 1700
1040 IF NOT INT (RND*10) THEN GO TO 6000
1050 INPUT "": PRINT ';"Cosa vuoi fare?"''"(1) combatti (2) scappi"
1060 PAUSE 0: IF INKEY$<"1" OR INKEY$>"2" THEN GO TO 1060
1070 IF INKEY$="2" THEN GO TO 8500
1080 IF INKEY$="1" THEN GO TO 3000
1100 IF tr THEN PAUSE 50: RETURN
1110 LET cn=INT (RND*(500/(INT (csh/500)+1)))+50
1120 PRINT INK 6;"Hai trovato ";cn;" monete d'oro."
1130 LET csh=csh+cn
1140 IF csh>=tres THEN GO TO 2500
1150 GO TO 4100
1200 IF INT (RND*2) THEN PAUSE 50: RETURN
1210 PRINT '"Hai trovato una chiave."
1220 LET ky=ky+1: GO TO 4100
1300 PRINT '"Non c'e' nulla!";: PAUSE 50: GO TO 8000
1400 IF INT (RND*2) THEN RETURN
1410 PRINT '"Hai trovato del vino e del cibo."''"Riposati."
1420 LET str=str+INT (RND*50)+5: IF str>500 THEN LET str=500
1430 PRINT '" ZZZZZ ZZZZZ ZZZZZ ZZZZZ ZZZZZ"
1440 FOR f=1 TO 5: BEEP 1.5,-48: BEEP 0.2,47: PAUSE 25: NEXT f
1450 PRINT ''"E' ora di andare...";: PAUSE 70
1460 GO TO 8000
1500 PRINT '"La porta e'chiusa!";: GO SUB 4500: PRINT "Hai la chiave?";
1510 PAUSE 50: IF ky<1 THEN GO TO 6500
1520 PRINT ' FLASH 1;"Si.";: PAUSE 20: PRINT "Vuoi usarla?(s/n)"
1530 PAUSE 0: IF INKEY$<>"n" AND INKEY$<>"N" AND INKEY$<>"s" AND INKEY$<>"S" THEN GO TO 1530
1540 IF INKEY$="n" OR INKEY$="N" THEN GO TO 6500
1550 PRINT '"Si....Apro la porta.";: LET ky=ky-1: PAUSE 20: PRINT "....click"
```

```
1600 PRINT '" La porta si apre."
 ;
1610 GO SUB 4100
1620 PRINT "......guarda!"
1630 IF NOT fl AND tr THEN
  GO TO 1800
1640 PAUSE 50
1650 GO TO 1000+100*INT (RND*5)
1700 CLS : GO SUB 3500
1710 PRINT 'TAB 6;"LA TRISTE CON
 CLUSIONE"'TAB 6;"**********
 **********"
1720 PRINT '"Sei finito nelle m
 ani di WARLOCK"''"Egli ti h
 a trasformato in ";k$(INT (
 RND*3)+1)
1740 PRINT '"I tuoi punti-coragg
 io erano ";hr
1750 PRINT ' INK 6;"Hai trovato
 ";csh;" monete d'oro."
1760 GO SUB 4000
1770 GO TO 1900
1800 GO SUB 3500
1810 PRINT  FLASH 1; INK 6;'AT 5
 ,10;"COMPLIMENTI!": FLASH 0
 : INK 7
1820 PRINT '"      Hai ingannato
  Warlock"
1830 PRINT 'TAB 8;"ed i suoi mos
 tri."
1840 FOR i=1 TO 3: GO SUB 4100:
 NEXT i
1850 PRINT ' INK 6;"  Hai trovat
 o ";csh;" monete d'oro."
1860 PRINT ';"  Hai trovato ";ky
 ;(" chiave." AND ky=1)+(" c
 hiavi." AND ky<>1)
1870 PRINT '"  Hai sconfitto ";d
 ed;(" Mostri." AND ded>1)+(
 " Mostro." AND ded<=1)
1880 PRINT '"I tuoi punti-coragg
 io ";("sono:" AND str>0);("
 erano:" AND str<=0);hr
1900 IF hr>hi THEN  LET hi=hr
1910 PRINT AT 2,20;"Record: ";hi
1920 INPUT "": PRINT #0;TAB 3;"U
 n' altra avventura?(s/n)":
 PAUSE 0
1930 IF INKEY$="s" OR INKEY$="S"
 THEN  CLS : GO TO 50
1940 IF INKEY$="n" OR INKEY$="N"
 THEN  GO SUB 4100: CLS :
 STOP
1950 GO TO 1920
2000 GO SUB 3500
2010 PRINT '"   Non hai piu' forz
 a."
2020 PRINT '"   Warlock ha vinto
 ancora!"
2030 GO SUB 4000
2040 PRINT 'TAB 8;"Prima di mori
 re"
2050 GO TO 1850
2500 GO SUB 4100: CLS : PRINT
  FLASH 1; INK 6;AT 1,10;"BE
 N FATTO!": FLASH 0: INK 7
2510 PRINT ''TAB 8;"Hai trovato
 il"
2520 PRINT 'TAB 6;"tesoro di War
 lock!"
2525 PRINT 'TAB 1;"Ora torna fuo
 ri";: PAUSE 80: PRINT "..se
  ci riesci!"
2530 PRINT ''TAB 3;"Sei al livel
 lo ";fl
2540 PRINT 'TAB 3;"Per uscire de
 vi"';TAB 3;"attraversare un
 a porta"
2545 PRINT TAB 3;"posta al livel
 lo 0."
2560 PAUSE 50: PRINT '''TAB 10;"
 Buona fortuna!"
2570 LET tr=1: GO TO 5500
3000 LET ht=INT (RND*10)+1
3010 LET str=str-ht*(INT (RND*2)
 +1): IF str<1 THEN  GO TO 2
 000
3020 IF ht>m(mn) THEN  GO TO 305
 0
3030 PRINT '"Non hai colpito ";m
 $(mn)
3040 GO SUB 4500: GO TO 1050
3050 LET blo=ht-m(mn)
3060 PRINT '(" Hai ferito "
  AND blo>=3);(" Hai spavent
 ato " AND blo<=2);m$(mn):
  GO SUB 4100
3070 LET n(mn)=n(mn)-blo
3080 IF n(mn)<1 THEN  GO TO 3110
3090 IF n(mn)<3 THEN  PRINT ';m$
 (mn)( TO g(mn));" sembra mo
 lto stanco."
3100 GO TO 1050
3110 PAUSE 50: PRINT ';m$(mn)(
  TO g(mn));" scappa.":
  GO SUB 4100: PRINT '" Ma l
 a tua forza e' ora ";str
3120 LET hq=hq+g(mn)*(INT (RND*3
 )+1): LET ded=ded+1
```

```
3130 GO TO 1100+100*INT (RND*2)
3500 CLS : PRINT AT 1,12;"WARLOC
    K"'TAB 12;"*******":
    RETURN
4000 FOR f=1 TO 11: BEEP z(f),y(
    f): NEXT f: RETURN
4100 FOR f=10 TO 20:: FOR h=0
    TO 3: BORDER h: BEEP .016,
    f*2-h: NEXT h: NEXT f:
    RETURN
4500 FOR f=5 TO 6: BEEP .5,-6*f:
    NEXT f: PAUSE 50: RETURN
5000 GO SUB 3500
5010 PRINT AT 15,6;"Seguono le i
    struzioni"
5020 GO SUB 5500: GO SUB 3500
5030 DIM m$(9,15): DIM a$(3,12)
5040 DIM m(9): DIM n(9): DIM g(1
    5)
5050 DIM k$(3,6): DIM z(11):
    DIM y(11)
5060 RESTORE : FOR f=1 TO 9:
    READ m$(f): READ m(f):
    READ n(f): READ g(f):
    NEXT f
5070 FOR f=1 TO 3: READ a$(f):
    NEXT f
5080 FOR f=1 TO 11: READ z(f):
    READ y(f): NEXT f
5090 FOR f=1 TO 3: READ k$(f):
    NEXT f
5100 PRINT '"Sei un giovane eroe
    all' attaccodel Palazzo di
    Warlock."
5110 PRINT '"Stai cercando di tr
    ovare il suo ";TAB 11;
    FLASH 1; PAPER 6;"TESORO."
5120 PRINT '"...Warlock non e' n
    el palazzo..."''TAB 13;
    FLASH 1;"MA"; FLASH 0
5130 PRINT '" c'e chi fa una buo
    na guardia!"
5140 PRINT '" ATTENTI AL RITORN
    O DI WARLOCK!"
5150 FOR f=1 TO 3: FOR i=4 TO 7:
    FOR h=2 TO 6: BEEP .01,f+i
    +h: BORDER h: BEEP .01,i+h+
    f: BORDER i: BEEP .01,h+f-i
    : BORDER f: NEXT h: NEXT i:
    NEXT f: BORDER 0
5160 LET hr=0: LET tr=0: LET ky=
    0
5170 LET tres=1500+100*INT (RND*
    10)
5180 LET fl=0: LET ded=0: LET hq
    =0
5190 LET str=500: LET csh=0
5500 PRINT AT 21,1;"Premi un tas
    to per continuare.":
    PAUSE 0: CLS : RETURN
6000 PRINT '" Qualcuno ti colpis
    ce e"'"ti porta via dell'o
    ro."
6010 LET csh=csh-INT (csh/(INT (
    RND*3)+1))
6020 PRINT ''; INK 6;" Ora hai "
    ;csh;" monete d'oro."
6030 IF tr AND csh<tres THEN
    LET tr=0: PRINT ''TAB 3;"
    Ricomincia a cercare."
6050 GO TO 4500
6500 PRINT ' FLASH 1;"No!";:
    PAUSE 50: GO TO 8000
7500 DATA "lo Spirito",5,8,10,"i
    l Gigante",3,10,10,"il Demo
    nio",2,10,10,"il Vampiro",2
    ,8,10,"il Lupo Mannaro",4,9
    ,15,"l'Ossesso",3,4,9,"il F
    antasma",2,8,11,"il Drago",
    4,8,8,"WARLOCK",8,12,7
7600 DATA "un Corridoio.","una P
    orta.","una Scala."
7700 DATA 1,4,1,4,0.3,4,1.2,4,0.
    75,7,0.5,6,1,6,0.3,4,0.7,4,
    0.5,3,1,4
7800 DATA "lepre","rana","pulce"
8200 IF NOT INT (RND*5) THEN
    GO TO 1000
8210 RETURN
8500 PRINT '" Codardo!": LET hq=
    hq-g(mn)*(INT (RND*3)+1):
    IF hq<1 THEN LET hq=0
8510 GO TO 8520+10*INT (RND*3)
8520 PRINT '" Sfortunatamente, n
    ella fretta"'" hai perso d
    ell' oro.": GO TO 6010
8530 GO TO 4500
8540 PRINT '" Che sfortuna!"'"T
    i ha intrappolato.":
    GO SUB 4500: GO TO 1050
9000 CLEAR : LET S$="WARLOCK"
9010 SAVE S$ LINE 0: BEEP 1,10
9020 PRINT "RIAVVOLGI IL NASTRO
    E PREMI UN"'"TASTO PER LA V
    ERIFICA"
9030 PAUSE 0
9040 VERIFY S$: BEEP 1,0
9050 STOP
```

La valle della morte

di **J. K. Laskowshi**
trad. e adatt.
di **Angelo Motta**

Vivrete con lo ZX81 l'intensità drammatica di un assalto disperato



Lo spunto di questo gioco di strategia è stato preso dalla battaglia di Balaklava (25 ott. 1854) durante la guerra di Crimea, nella quale in generale inglese Lord Cardingam, alla testa di un pugno di uomini, tentò di contrastare uno schieramento di ben 30 cannoni russi nella valle della morte.
Lanciato il programma, viene visualizzato sullo schermo una scacchiera di 10 × 10, rappresentante appunto "La valle della morte". Ai lati sinistro, destro ed inferiore della stessa sono situati i 30 cannoni russi con un valore da 1 a 9 assegnato. Il vostro plotone di uomini è posto al centro del lato superiore, pronto ad entrare nella valle.
La prima mossa che dovrete effettuare sarà quella di entrare nella valle e dovrete pertanto inserire D per destra, A per avanti o S per sinistra. Non appena entrati avrà inizio il gioco vero e proprio: Il nemico ha a disposizione 3 colpi (uno per lato) mentre voi ne avete uno solo. La potenza del colpo è determinata dal valore del cannone o da quello assegnato al vostro uomo. Per sparare dovete inserire prima la lettera corrispondente all'uomo della vostra brigata che intendete far sparare (lettera dalla A alla I come da schema che lo ZX81 fa apparire sulla destra del video), quindi inserire la direzione dello sparo indicato (D - A - S). Quest'ultima lettera determinerà inoltre la direzione di avanzamento nella valle.
Una volta raggiunto il fondo della valle inizierà la ritirata. Per ogni cannone colpito vi verrà assegnato il corrispondente punteggio e per ogni vostro uomo abbattuto verrà detratto il rispettivo valore.
Scopo del gioco è cercare di riuscire ad ottenere alla fine della gara almeno un punteggio superiore allo zero, che è già da considerarsi un buon risultato. A video viene riportato il record raggiunto in tale gioco (da un fenomeno, penso) di 63 punti; dopo diverse partite di prova effettuate superare i 20 punti è stato notevolmente difficile.

La valle della morte versione per ZX81 16K

```
  5 CLS
 10 DIM A(3,11)
 20 DIM B(3,3)
 30 DIM C$(32)
 50 RAND 0
 60 FAST
 70 PRINT TAB 5;"LA VALLE DELLA
MORTE"
 80 LET MOSSE=0
 90 LET FORZA=0
100 LET F=0
110 LET M=1
120 LET N=6
130 LET X=2
140 LET TOT=0
160 LET WR=63
170 PRINT AT 6,22;"THE LIGHT";T
AB 23;"BRIGADE",,,TAB 25;"ABC";T
AB 25;"DEF";TAB 25;"GHI";AT 18,2
2;"RECORD=";WR
180 LET B$=""
```

```
190 FOR I=1 TO 3
200 FOR J=1 TO 11
210 LET A(I,J)=INT (RND*9+1)
220 GOTO 230+(3-I)*20
230 PRINT AT 4+J,8;A(1,J);"..........";A(3,J)
240 GOTO 270
250 LET B$=B$+CHR$ (28+A(2,J))
270 NEXT J
280 NEXT I
290 PRINT TAB 9;B$
300 GOSUB 1200
310 SLOW
320 LET MOSSE=MOSSE+1
330 PRINT AT 21,0;"SPARA: A - I DIREZ.: S - A - D"
340 IF MOSSE=1 THEN PRINT AT 21,0;"AVANZA NELLA VALLE "
350 INPUT B$
360 PRINT AT 20,0;C$;TAB 0;C$
370 IF NOT ((LEN B$=2 AND MOSSE>1) OR (LEN B$=1 AND MOSSE=1)) THEN GOTO 330
380 LET B$=B$+B$
385 LET X=0
390 LET X=(CODE B$(2)=56)+(2 AND CODE B$(2)=38)+(3 AND CODE B$(2)=41)
400 IF X=0 THEN GOTO 330
410 IF MOSSE=1 THEN GOTO 550
420 IF CODE B$<38 OR CODE B$>46 THEN GOTO 330
430 LET G=INT ((CODE B$-37.5)/3)+1
440 LET H=CODE B$-37-3*(G-1)
450 IF B(G,H)<1 THEN GOTO 330
460 LET SPARO=-1
470 LET D$="*"
480 LET RANGE=B(G,H)
490 LET MI=M+G-2
500 LET NI=N+H-2
505 IF X<>2 AND MI<1 THEN GOTO 330
510 GOSUB 1400
520 IF SPARO=-1.1 THEN GOTO 330
530 IF PUNTI>0 THEN PRINT AT 21,0;"BUON COLPO.  PUNTEGGIO ";PUNTI
540 GOSUB 1900
550 LET F=SGN (9.5-MOSSE)
560 LET M=M+F
570 LET N=N+X-2
580 IF N>1 AND N<11 THEN GOTO 610
590 LET N=N-X+2
600 LET X=2
610 GOSUB 1200
620 LET O1=B(3,1)+B(3,2)+B(3,3)
630 LET O2=O1+B(2,1)+B(2,2)+B(2,3)
640 IF MOSSE=20 OR (MOSSE=19 AND O1<=-3) OR (MOSSE=18 AND O2<=-6) THEN GOTO 2000
650 FOR X=1 TO 3
660 PRINT AT 21,0;C$
670 LET SPARO=1
680 LET R=3
690 IF MOSSE>17 THEN LET R=20-MOSSE
695 LET O=N+1
700 LET P=N
705 IF X<>2 THEN LET O=M+1
710 IF X<>2 THEN LET P=M+3-R
720 FOR V=P-1 TO O
730 IF A(X,V)<>0 THEN GOTO 770
740 NEXT V
750 NEXT X
760 GOTO 320
765 IF X<>2 THEN LET P=M
770 IF X=2 THEN LET R=3
775 LET CANNONI=INT (RND*R+2-R)
780 IF A(X,P+CANNONI)=0 THEN GOTO 770
790 LET MI=P+CANNONI
800 LET RANGE=A(X,MI)
810 LET NI=MI
820 IF X=2 THEN LET MI=12
830 IF X<>2 THEN LET NI=(X-1)*6
840 LET D$="X"
850 PRINT AT 8,0;"SPARO ";X
860 GOSUB 1400
870 IF PUNTI<0 THEN PRINT AT 21,0;"COLPITO - PERDI ";ABS PUNTI;AT 8+G,24+H;" "
880 GOSUB 1900
890 IF FORZA<=0 THEN GOTO 2000
900 GOTO 750
1200 LET Z=N+7-X+2
1210 DIM C(11,11)
1260 PRINT AT 3+M-F,Z;"...";TAB Z;"...";TAB Z;"...";AT 3,Z;"   ";TAB Z;"   "
1270 FOR G=1 TO 3
1280 FOR H=1 TO 3
1290 IF MOSSE>0 THEN GOTO 1320
1300 LET B(G,H)=INT (RND*9+1)
1310 LET FORZA=FORZA+B(G,H)
1320 IF MOSSE>17 AND B(G,H)=-1 AND M+G-2<1 THEN LET B(G,H)=-28
1330 PRINT AT 2+M+G,6+N+H;CHR$ (28+B(G,H))
1340 IF M+G-2>0 THEN LET C(M+G-2,N+H-2)=B(G,H)
1350 NEXT H
1360 NEXT G
1370 RETURN
1400 LET PUNTI=0
1410 FOR S=1 TO RANGE
1420 LET T=-S*SPARO
1430 LET U=S*(2-X)*SPARO
1435 IF X<>2 THEN LET T=0
1440 PRINT AT MI+T+4,NI+U+8;D$
1450 IF NI+U<1 OR NI+U>11 OR MI+T>11 THEN GOTO 1490
1460 IF C(MI+T,NI+U)>0 THEN GOTO 1550
1470 NEXT S
1480 RETURN
1490 LET RANGE=S-1
1500 LET Y=MI
1510 IF X=2 THEN LET Y=NI
1520 LET PUNTI=A(X,Y)
1530 LET A(X,Y)=0
1540 RETURN
1550 IF SPARO=-1 THEN GOTO 1650
1570 LET RANGE=S
1580 LET G=MI+T-M+2
1590 LET H=NI+U-N+2
1600 LET PUNTI=-B(G,H)
1610 LET FORZA=FORZA+PUNTI
1620 LET B(G,H)=-1
1630 LET C(M+G-2,N+H-2)=0
1640 RETURN
1650 LET SPRO=-1.1
1655 PRINT AT 21,0;"SPARO ILLEGALE - PASSI IL TURNO"
1660 GOSUB 1200
1670 RETURN
1900 LET TOT=TOT+PUNTI
1910 PRINT AT 18,0;"PUNTEGGIO ";TOT;"    "
1920 IF RANGE=0 THEN RETURN
1930 LET D$="."
1935 PRINT AT 8,0;"  "
1940 GOSUB 1400
1950 RETURN
```

La valle della morte




```
2000 PRINT AT 19,0;"FINE DELLA B
ATTAGLIA";TAB 0;"NESSUN SOPRAVVI
SSUTO"
2010 IF FORZA>0 THEN PRINT AT 20
,0;"VIVI ";FORZA;" AGG.TI AL PUN
TEGGIO"
2020 LET PUNTI=FORZA
2030 GOSUB 1900
2040 PRINT AT 21,0;"IL RECORD RI
MANE INVARIATO"
2050 IF TOT>WR THEN PRINT AT 21,
0;"HAI BATTUTO IL RECORD - BRAVO
"
2060 FOR I=1 TO 50
2070 NEXT I
2080 PRINT AT 0,0;"[redacted]"
2090 IF INKEY$="S" THEN RUN
2100 IF INKEY$="" THEN GOTO 2090
2110 STOP
2500 SAVE "VALLEMORTE"
2510 RUN
```

E tu cosa aspetti
a spedire il tuo
programma?

J.SOFT: OBIETTIVO





Corsa nel tunnel

di **David Link**
trad. e adatt.
di **Marcello Spero**

Raggiungere lo spazioporto sarà per la vostra astronave un impresa veramente difficile!



Vagamente ispirato ad uno dei più popolari "arcade", Tunnel run vi porta ad esplorare con la vostra astronave un labirinto spaziale. Il gioco si compone di quattro fasi, durante le quali dovrete attraversare il tunnel dei missili, il campo di asteroidi, il tunnel contorto, per portare infine il vostro veicolo ad un perfetto atterraggio sulla piazzola dello spazioporto.

Il programma basa il suo funzionamento sul fatto che dando alla variabile di sistema SCRCT, all'indirizzo 23692, valori compresi fra uno e 255, lo schermo scorrerà verso l'alto per un numero corrispondente di volte senza che vi venga posta l'infernale domanda "scroll?".

I tasti usati per muoversi sono sempre il "5" per spostarsi a sinistra, e l'"8" per spostarsi a destra. In aggiunta a questi, durante il secondo stadio del gioco il tasto "0" serve a fare fuoco, per distruggere l'astronave blu che appare di tanto in tanto. Attenti, però: ogni volta che userete il vostro laser, perderete 20 punti; in caso di centro, comunque, ne guadagnerete 100.

Nell'ultima fase di gioco dovrete cercare di atterrare sulla piazzola gialla. In questo stadio avrete a disposizione il comando "R", per azionare i retrorazzi, oltre naturalmente, ai due tasti di direzione.

Se vi dovesse capitare, alla fine del gioco, di fermare accidentalmente il programma, fatelo ripartire con GO TO 45. Dando RUN, infatti, perdereste la registrazione del precedente record, proprio come se aveste appena caricato il gioco.

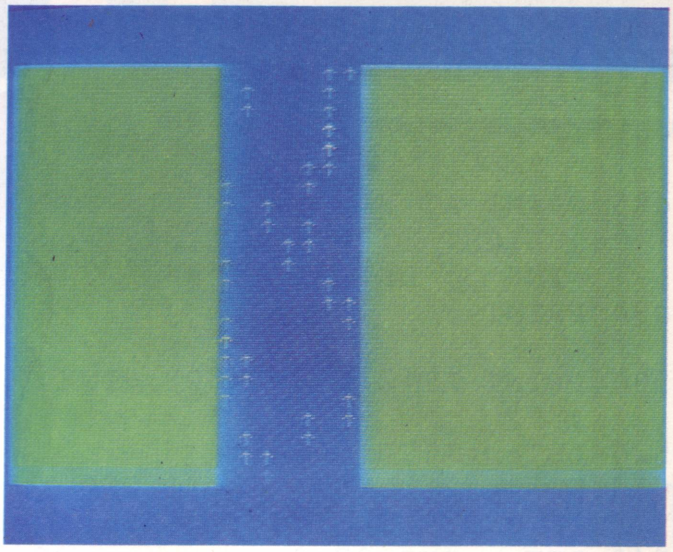
Per quanto riguarda il lavoro di trascrizione, non ci sono particolari problemi; il programma, infatti, non è né lungo né complesso, adattandosi ugualmente bene al 16K come al 48K.

Detto questo, a voi i comandi e buon divertimento!

Corsa nel tunnel
versione per ZX Spectrum 16K/48K

```
 1 FOR f=0 TO 7
 2 READ a: POKE USR "L"+f,a
 3 NEXT f
 4 DATA BIN 00000000,BIN 010
   00010,BIN 10011001,BIN 10
   111101,BIN 11111111,BIN 1
   0011001,BIN 01000010,
   BIN 00000000
10 FOR f=0 TO 7
20 READ a: POKE USR "P"+f,a
30 NEXT f
40 DATA BIN 01111110,BIN 001
   11100,BIN 00011000,BIN 01
   111110,BIN 01111110,BIN 0
   1111110,BIN 01011010,
   BIN 00011000
41 LET hi=0: LET z=0
```

```
42 BORDER 6: PAPER 6: INK 1:
   CLS
43 PRINT "     T U N N E L
   R U N"''''"Sei stato sfidat
   o ad esplorare un compless
   o labirinto spaziale Rientr
   erai vivo alla base?"''"Il
   labirinto e' composto da
     quattro sezioni:"'"1> Tun
   nel dei missili"'"2> Astero
   idi (tasto '0' = fuoco)"'"3
   > Tunnel contorto"'"4> Piaz
   zola di atterraggio"'"    (t
   asto 'R' = retrorazzi,
     atterra sulla zona gial
   la)"
45 PRINT ''"PREMI UN TASTO PER
    INIZIARE!"
46 IF INKEY$<>"" THEN
    GO TO 46
48 PAUSE 0
50 LET a=15: LET sc=0
70 BORDER 1: PAPER 1: INK 0:
   CLS
80 PRINT AT 21,0;"{10SG8}
     {15SG8}"
90 FOR f=1 TO 200
95 IF a<=9 OR a>=17 THEN
    GO TO 900
100 PRINT AT 0,a;"P"
110 POKE 23692,255
112 LET a=a+(INKEY$="8")-(
    INKEY$="5")
117 PRINT AT 21,0;
120 PRINT AT 21,INT (RND*7)+10;
     INK 6;"↑"
130 PRINT  INK 4;"{10SG8}
     {15SG8}"
140 LET b=ATTR (0,a)
150 IF b<>8 AND b<>12
     THEN  GO TO 900
155 LET sc=sc+1
157 BEEP .005,0: BEEP .005,-2
160 NEXT f
170 FOR f=1 TO 22
180 PRINT AT 0,a;"P"
190 LET a=a+(INKEY$="8")-(
    INKEY$="5")
195 POKE 23692,255
200 PRINT AT 21,0;"
                              "
210 PRINT "
                              "
```



```
220 LET b=ATTR (0,a)
230 IF b<>8 AND b<>12
     THEN  GO TO 900
240 LET sc=sc+1
250 BEEP .01,0: BEEP .01,-2
260 NEXT f
270 PRINT AT 0,a;"P"
280 PRINT AT 10,10; FLASH 1;"Pu
    nti ";sc
290 FOR f=1 TO 5
300 FOR n=10 TO 20
310 BEEP .1,n
320 NEXT n
330 NEXT f
340 PAUSE 250
342 BORDER 2: PAPER 2: CLS
343 PRINT AT 0,a;"P"
345 FOR f=1 TO 3
350 BEEP .2,-2: BEEP .3,-2:
     BEEP .5,3.2
360 NEXT f
370 BEEP 1,3.2
375 LET c=20
380 FOR f=1 TO 200
390 PRINT AT 0,a;"P"
400 LET a=a+(INKEY$="8")-(
    INKEY$="5")
410 IF INKEY$<>"0" THEN
     GO TO 480
415 LET sc=sc-20
420 FOR n=1 TO 21
430 PRINT AT n,a; INK 6;"|"
435 NEXT n
440 IF a<>c THEN  GO TO 470
450 LET sc=sc+100
460 FOR x=1 TO 30: PRINT AT 21,
    c; INK 6; PAPER 0; FLASH 1;
    "#": NEXT x
```

```
470 FOR n=1 TO 21: PRINT AT n,a
  ;" ": NEXT n
480 POKE 23692,255
481 IF z=5 THEN  PRINT AT 17,c;
  " "
485 PRINT AT 21,0; INK 3;
   BRIGHT 1;"###";AT 21,31;
   INK 3; BRIGHT 1;"#"
490 PRINT "
                           "
500 PRINT AT 21,(RND*30); INK 4
  ;"©"
520 LET b=ATTR (0,a)
530 IF b<>16 THEN  GO TO 900
535 IF z<5 THEN  GO TO 559
540 LET c=RND*27+3
555 BEEP .005,0: BEEP .005,-2
557 PRINT AT 21,c; INK 1;"L"
558 LET z=0
559 LET z=z+1
560 NEXT f
570 FOR f=1 TO 22: PRINT AT 21,
  0;"
                ": NEXT f
575 BORDER 3: PAPER 3: CLS
580 PRINT AT 0,a;"P"
590 PRINT AT 10,10; FLASH 1;"Pu
  nti ";sc
600 BEEP 1,9: BEEP 1,10.7:
   BEEP 1,9: PAUSE 130:
   BEEP 2,0
605 LET tu=13
607 PRINT AT 10,10;"
  "
610 FOR f=1 TO 172
620 PRINT AT 0,a;"P"
630 LET a=a+(INKEY$="8")-(
  INKEY$="5")
640 POKE 23692,255
650 LET tu=tu+(RND>.5)-(RND<.5)

660 LET tu=tu+(tu<1)-(tu>24)
670 PRINT AT 21,tu; PAPER 3;
   INK 0;"    "
675 IF RND>.7 THEN  PRINT AT 21
  ,(RND*5)+tu; INK 1;"L"
680 PRINT  INK 7; BRIGHT 1;"
  {32SG8}"
685 IF f>149 THEN  PRINT AT 21,
  0; PAPER 3; INK 0;"
                              "
690 LET b=ATTR (0,a)
700 IF b<>24 THEN  GO TO 900
705 LET sc=sc+1
710 BEEP .005,0: BEEP .005,-2
720 NEXT f
730 BORDER 0: PAPER 0: INK 7:
  CLS
732 PRINT AT 0,a;"P"
734 LET g=INT (RND*30)+1
735 PRINT AT 15,0; INK 1;"#####
  ###########################
  "
736 FOR f=16 TO 21: PRINT AT f,
  0; INK 2;"{32SG8}": NEXT f
737 PRINT AT 15,g; INK 6;"#"
738 LET x=0
739 PAUSE 150: LET ox=0: LET oa
  =a
740 PRINT AT ox,oa;" ";AT x,a;"
  P"
745 LET oa=a: LET ox=x
750 LET a=a+(INKEY$="8")-(
  INKEY$="5")
755 LET a=a+(a<0)-(a>31)
760 LET x=x+1
765 PAUSE 5
770 IF INKEY$="r" THEN  LET x=x
  -2*(x>1)
780 IF x>=15 THEN  GO TO 800
785 BEEP .005,0: BEEP .005,-2
790 PRINT AT x,a;" ": GO TO 740

800 IF a<>g THEN  GO TO 900
810 PAPER 0: INK 7: FLASH 1:
  CLS
830 PRINT AT 5,10;"PUNTI ";sc
832 IF sc>hi THEN  LET hi=sc
833 PRINT AT 8,10;"RECORD ";hi
835 CIRCLE 125,85,80
840 FOR f=1 TO 20: BEEP .01,f:
  NEXT f
850 GO TO 920
900 CLS : INK 6: PRINT AT 10,0;
  "Sei stato disrutto"'''"Hai
  totalizzato ";sc;" punti"

915 FOR f=0 TO -10 STEP -1:
   BEEP .01,f: NEXT f
920 PRINT : PRINT "Premi s per
  un altra partita,"'''"n per
  smettere"
930 IF INKEY$="" THEN  GO TO 93
  0
935 FLASH 0: CLS
948 IF INKEY$="s" THEN  GO TO 4
  5
```

Disassembler Z80

di **Massimo Magnani**

Un utile programma disassemblatore per lo Spectrum

Coloro che sanno programmare in linguaggio macchina troveranno senz'altro utile poter controllare facilmente i loro programmi esaminando il disassemblato del codice oggetto e anche coloro che desiderano avvicinarsi a questo meraviglioso modo di programmare potranno trovare numerosi spunti dal tradurre facilmente le numerose routine che si trovano in quasi tutti i programmi commerciali.

Il programma è scritto interamente in BASIC e sta nei primi 16Kbyte di RAM. Per rendere il programma più veloce sono stati usati i numeri di linea più bassi per le routines usate di frequente, comunque la linea 1 permette di far partire il programma col solito RUN.

All'inizio bisogna scegliere fra le due opzioni DUMP e DIZ80: con la prima si ottiene in uscita un lista (in esadecimale) dei valori in memoria ed il loro equivalente simbolo ASCII (utile per le tabelle), con la seconda si entra nel disassemblatore vero e proprio. Alla richiesta degli indirizzi si può rispondere sia con numeri decimali che con esadecimali, nel secondo caso è necessario far seguire al numero la lettera "h" (non ha importanza se le lettere eventualmente presenti nel numero sono maiuscole o minuscole).

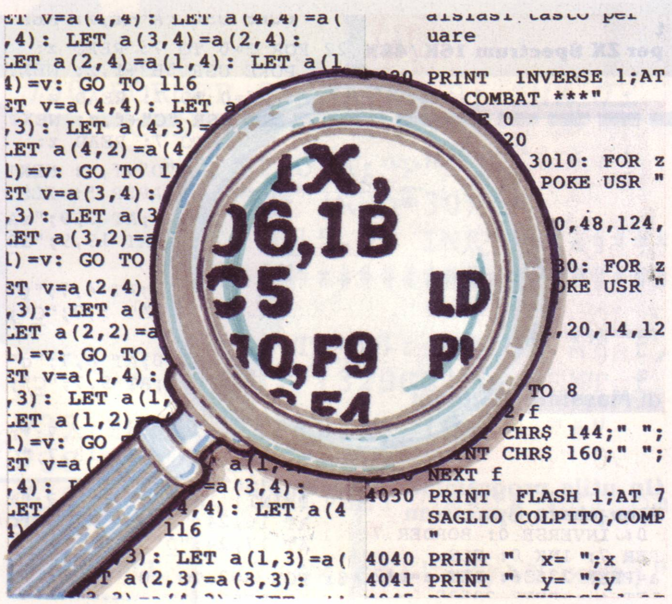
Usando il disassemblatore si può decidere di far fermare l'elaborazione in corrispondenza dei codici di fine routine e di salto incondizionato, in questo caso se si desidera proseguire col disassemblato occorre digitare il comando CONTINUE.

Descrizione del programma

Linee	Commento
20	Funzione che traduce in stringa esadecimale il valore dato come argomento.
50-100	Riportano in n il contenuto della locazione corrente ed in n$ il corrispondente valore esadecimale.
150	Prende il primo codice operativo in op e lo stampa.
200	Riporta in h$ il valore di d come numero esadecimale di 2 byte.
900	Riporta in x e y il valore dei bit 3-5 e 0-2 di op.
1000-1020	Vari modi di indirizzamento.
1100-1120	Scompongono x$ alla 1ª parentesi.
1500	Trova il registro interessato.
1600	Trova la giusta condizione.
1700	Trova operazione aritmetica o logica.
1800	Trova operazioni di shift o rotazione.
4000-4500	Data.
6000-6090	Programma DUMP.
6500-6540	Assegnazione indirizzi e opzione stampante.
6000-6660	Traduce se necessario il numero esadecimale in decimale.
7000-7005	Scelta programma (DUMP-DIZ80).
7010	Inizio DISASSEMBLER.
7015-7024	Inizializzazione.
7030-7180	Corpo principale.
7200-7240	Controllano se è finita l'elaborazione, se no, passa alla prossima istruzione.
7260-7340	Se il codice operativo è CBh.
7400-7827	Se c'è indirizzamento indicizzato.
7900-8255	Se il codice operativo è EDh.
9000	Codice non riconosciuto.

Disassembler Z80 versione per ZX Spectrum 16K/48K

```
   1 GO TO 7000
  20 DEF FN q$(n)=CHR$ (48+INT (
     n/16)+7*(n>=160))+CHR$ (48+
     n-INT (n/16)*16+7*(n-INT (n
     /16)*16>=10))
  49 REM *** LEGGE BYTE ***
  50 LET n=PEEK an
 100 LET n$=FN q$(n): RETURN
 150 LET an=an+1: GO SUB 50:
     PRINT n$;: LET op=n:
     RETURN
 199 REM * H$=D in esadecimale *
 200 LET j=INT (d/256): LET h$=
     FN q$(j): LET j$=FN q$(d-j*
     256): LET h$=h$+j$:
     RETURN
 900 LET x=INT ((op-k)/8): LET y
     =op-k-x*8: RETURN
```



```
1000 PRINT TAB ts;x$: GO TO 7200
1001 GO TO 1010
1002 GO SUB 150: LET aa=n: LET y
   $=n$: GO SUB 150: PRINT
   TAB ts;x$;n$;y$;"h";TAB td;
   aa+n*256: GO TO 7200
1003 GO SUB 150: LET j$=n$:
    LET aa=n: GO SUB 150:
    GO SUB 1100: PRINT TAB ts;
   x$;n$;j$;"h";y$;TAB td;aa+2
   56*n: GO TO 7200
1004 GO SUB 150: PRINT TAB ts;x$
   ;n$;"h": GO TO 7200
1005 GO SUB 150: LET x$=x$+"("+z
   $+"+"+n$+"h)": RETURN
1006 GO SUB 150: GO SUB 1100:
    PRINT TAB ts;x$;n$;"h";y$:
    GO TO 7200
1010 GO SUB 150: IF op>127
    THEN  LET op=op-256
1020 LET d=an+op+1: GO SUB 200:
    PRINT TAB ts;x$;h$;"h";
   TAB td;d: GO TO 7200
1100 LET b=0: FOR t=1 TO LEN x$
1110 IF x$(t)="(" THEN  LET b=t
1120 NEXT t: LET y$=x$(b+1 TO ):
    LET x$=x$( TO b): RETURN
1500 RESTORE 4200: FOR t=0 TO re
   g: READ y$: NEXT t: LET x$=
   x$+y$: RETURN
1600 RESTORE 4300: FOR t=0 TO x:
    READ y$: NEXT t: LET x$=x$
   +y$: RETURN
1700 RESTORE 4100: FOR t=0 TO x:
    READ x$: NEXT t: RETURN
1800 RESTORE 4400: FOR t=0 TO x:
    READ x$: NEXT t: RETURN
4000 DATA "0NOP","2LD BC,","0LD
   (BC),A","0INC BC","0INC B",
   "0DEC B","4LD B,","0RLCA","
   0EX AF,AF'","0ADD HL,BC","0
   LD A,(BC)","0DEC BC","0INC
   C","0DEC C","4LD C,","0RRCA
   ","1DJNZ ","2LD DE,"
4010 DATA "0LD (DE),A","0INC DE"
   ,"0INC D","0DEC D","4LD D,"
   ,"0RLA","1JR ","0ADD HL,DE"
   ,"0LD A,(DE)","0DEC DE","0I
   NC E","0DEC E","4LD E,","0R
   RA","1JR NZ,","2LD HL,","3L
   D (),HL","0INC HL","0INC H"
   ,"0DEC H","4LD H,","0DAA"
4020 DATA "1JR Z,","0ADD HL,HL",
   "3LD HL,()","0DEC HL","0INC
   L","0DEC L","4LD L,","0CPL
   ","1JR NC,","2LD SP,","3LD
   (),A","0INC SP","0INC (HL)"
   "0DEC (HL)","4LD (HL),","0
   SCF","1JR C,","0ADD HL,SP",
   "3LD A,()","0DEC SP","0INC
   A","0DEC A","4LD A,","0CCF"
4100 DATA "ADD A,","ADC A,","SUB
    ","SBC A,","AND ","XOR ","
   OR ","CP "
4200 DATA "B","C","D","E","H","L
   ","(HL)","A"
4300 DATA "NZ","Z","NC","C","PO"
   ,"PE","P","M"
4400 DATA "RLC ","RRC ","RL ","R
   R ","SLA ","SRA ","-","SRL
   "
4500 DATA "0POP BC","0RET","0POP
    DE","0EXX","0POP HL","0JP
   (HL)","0POP AF","0LD SP,HL"
   ,"2JP "," ","6OUT (),A","6I
   N A,()","0EX (SP),HL","0EX
   DE,HL","0DI","0EI","0PUSH B
   C","2CALL ","0PUSH DE"," ",
   "0PUSH HL"," ","0PUSH AF","
   "
6000 REM *** DUMP ***
6010 GO SUB 6500: LET an=a:
    PRINT
6030 LET x$="": LET d=an:
    GO SUB 200: PRINT h$;"- ";
6040 FOR y=1 TO 8: GO SUB 50:
    PRINT n$;" ";
```

Disassembler Z80

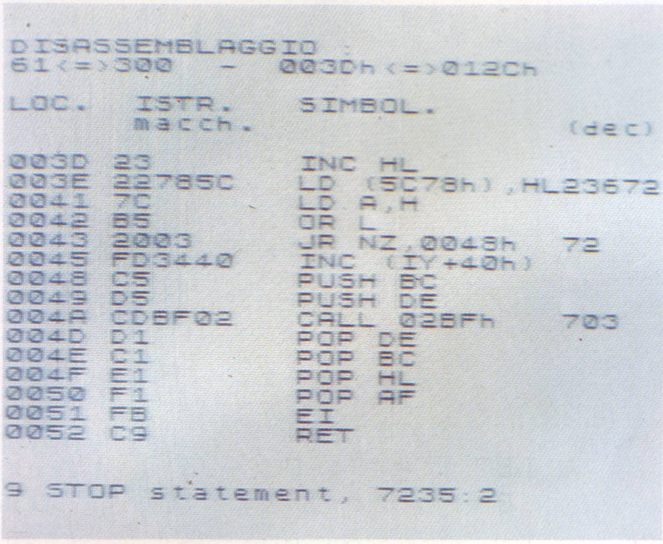
```
6050 IF n<32 OR n>127 THEN
   LET x$=x$+"?": GO TO 6080
6060 LET x$=x$+CHR$ n
6080 LET an=an+1: IF an>am
   THEN  PRINT TAB 38;x$:
   CLOSE #2: STOP : RUN
6090 NEXT y: PRINT TAB 38;x$:
   GO TO 6030
6500 INPUT "START ? "; LINE x$'"
   STOP ? "; LINE y$: GO SUB 6
   600: LET a=n: LET x$=y$:
   GO SUB 6600: LET am=n
6520 INPUT "OUTPUT ON PRINTER (s
   /n) ?";z$: LET pr=0: IF z$=
   "s" THEN  OPEN #2,"p":
   LET pr=1
6540 RETURN
6600 IF x$(LEN x$)<>"h" AND x$
   (LEN x$)<>"H" THEN  LET n
   =VAL x$: RETURN
6610 LET j=1: LET n=0: FOR t=
   LEN x$-1 TO 1 STEP -1:
   LET k=CODE x$(t)
6620 IF k<48 OR k>70 THEN
   GO TO 6650
6630 IF k>57 AND k<65 THEN
   GO TO 6660
6640 LET n=n+(k-48-(7 AND k>64))
   *j: LET j=j*16: NEXT t:
   RETURN
6650 IF k>96 AND k<103 THEN
   LET k=k-32: GO TO 6640
6660 PRINT #1;x$;" : ESADECIMALE
   ERRATO": PAUSE 0: RUN
6999 REM *** INIZIO **
7000 INPUT "1) DUMP      2) DIZ-8
   0"'"scegli ?";x$: IF x$="1"
   THEN  GO TO 6000
7005 IF x$<>"2" THEN  GO TO 70
   00
7009 REM *** DISASSEMBLER ***
7010 GO SUB 6500
7012 INPUT "Opzione JP/RET (s/n)
   ?";x$: LET opz=0: IF x$="s
   " OR x$="S" THEN  LET opz=1
7015 PRINT "DISASSEMBLAGGIO :"
   'a;"<=>";am;"  -";: LET d
   =a: GO SUB 200: PRINT h$;"h
   <=>";: LET d=am: GO SUB 200
   : PRINT h$;"h"
7020 LET pp=-1: LET cb=203:
   LET dd=221: LET ed=237:
   LET fd=253: LET ts=14:
   LET td=27: LET an=a
7024 PRINT '"LOC.   ISTR.";TAB ts
   ;"SIMBOL."'TAB 6;"macch.";
   TAB td;"(dec)"''
7030 LET d=an: GO SUB 200:
   PRINT h$;" ";
7040 GO SUB 50: PRINT n$;: LET o
   p=n: LET pp=op: IF op=cb
   THEN  GO TO 7260
7070 IF op=ed THEN  GO TO 7900
7080 IF op=dd THEN  LET z$="IX":
   GO TO 7400
7090 IF op=fd THEN  LET z$="IY":
   GO TO 7400
7100 IF op<64 THEN  RESTORE :
   FOR t=0 TO op: READ x$:
   NEXT t: LET d=VAL x$(1):
   LET x$=x$(2 TO ): GO TO 10
   00+d
7110 IF op>=128 THEN  GO TO 7140
7120 IF op=118 THEN  LET x$="HAL
   T": GO TO 1000
7130 LET x$="LD ": LET k=64:
   GO SUB 900: LET reg=x:
   GO SUB 1500: LET x$=x$+","
   : LET reg=y: GO SUB 1500:
   GO TO 1000
7140 IF op<192 THEN  LET k=128:
   GO SUB 900: GO SUB 1700:
   LET reg=y: GO SUB 1500:
   GO TO 1000
7150 LET k=192: GO SUB 900: IF y
   =0 THEN  LET x$="RET ":
   GO SUB 1600: GO TO 1000
7155 IF y=2 THEN  LET x$="JP ":
   GO SUB 1600: LET x$=x$+","
   : GO TO 1002
7160 IF y=4 THEN  LET x$="CALL "
   : GO SUB 1600: LET x$=x$+",
   ": GO TO 1002
7165 IF y=7 THEN  LET x$="RST "+
   (FN q$(x*8))+"h": GO TO 100
   0
7170 IF y=6 THEN  GO SUB 1700:
   GO TO 1004
7180 LET x=x+8*INT (y/2):
   RESTORE 4500: FOR t=0 TO x
```

```
    : READ x$: NEXT t: LET d=
    VAL x$(1): LET x$=x$(2 TO )
    : GO TO 1000+d
7190 REM *** CONTINUA ***
7200 IF pp=195 OR pp=24 OR pp=20
    1 OR pp=233 THEN  GO TO 723
    0
7205 LET an=an+1: IF an>am
    THEN  CLOSE #2: STOP :
    RUN
7210 GO TO 7030

7230 PRINT : IF opz=0 THEN
    GO TO 7205
7235 CLOSE #2: STOP : IF pr
    THEN  OPEN #2,"p"
7240 GO TO 7205
7250 REM *** OP=CBh ***
7260 GO SUB 150
7270 IF op>=64 THEN  GO TO 7300
7280 LET k=0: GO SUB 900: IF x=6
    THEN  GO TO 9000
7290 GO SUB 1800: GO TO 7340
7300 IF op<128 THEN  LET k=64:
    GO SUB 900: LET x$="BIT ":
    GO TO 7330
7310 IF op<192 THEN  LET k=128:
    GO SUB 900: LET x$="RES ":
    GO TO 7330
7320 LET k=192: GO SUB 900:
    LET x$="SET "
7330 LET x$=x$+STR$ x+","
7340 LET reg=y: GO SUB 1500:
    GO TO 1000
7390 REM *** INDICI ***
7400 GO SUB 150: IF op=cb THEN
    GO TO 7600
7410 IF op<64 THEN  GO TO 7550
7420 IF op>=128 THEN  GO TO 7500
7430 LET x$="LD ": LET k=64:
    GO SUB 900: IF y=6 THEN
    GO TO 7470
7440 IF x<>6 THEN  GO TO 9000
7450 GO SUB 1005: LET x$=x$+",":
    LET reg=y: GO SUB 1500:
    GO TO 1000
7470 IF x=6 THEN  GO TO 9000
7480 LET reg=x: GO SUB 1500:
    LET x$=x$+",": GO SUB 1005
    : GO TO 1000
7500 IF op>=192 THEN  GO TO 7550

7510 LET k=128: GO SUB 900: IF y
    <>6 THEN  GO TO 9000
7520 GO SUB 1700: LET x$=x$+",":
    GO SUB 1005: GO TO 1000
7550 IF op=33 THEN  LET x$="LD "
    +z$+",": GO TO 1002
7552 IF op=34 THEN  LET x$="LD (
    ),"+z$: GO TO 1003
7554 IF op=42 THEN  LET x$="LD "
    +z$+",()": GO TO 1003
7556 IF op=52 THEN  LET x$="INC
    ": GO SUB 1005: GO TO 1000
7558 IF op=53 THEN  LET x$="DEC
    ": GO SUB 1005: GO TO 1000
7560 IF op=54 THEN  LET x$="LD "
    : GO SUB 1005: LET x$=x$+",
    ": GO TO 1004
7570 IF op>192 THEN  LET op=op-1
    28
7580 LET x$="???": GO SUB 7700+o
    p: GO TO 1000
7600 LET x$="": GO SUB 1005:
    LET y$=x$: GO SUB 150:
    IF op>=64 THEN  GO TO 7630
7610 LET k=0: GO SUB 900: IF y
    <>6 OR x=6 THEN  GO TO 90
    00
7620 GO SUB 1800: LET x$=x$+y$:
    GO TO 1000
7630 IF op<128 THEN  LET k=64:
    LET x$="BIT ": GO TO 7660
7640 IF op<192 THEN  LET k=128:
    LET x$="RES ": GO TO 7660
7650 LET x$="SET ": LET k=192
7660 GO SUB 900: IF y<>6
    THEN  GO TO 9000
7670 LET x$=x$+STR$ x+","+y$:
    GO TO 1000
7709 LET x$="ADD "+z$+",BC"
7724 RETURN
7725 LET x$="ADD "+z$+",DE"
7734 RETURN
7735 LET x$="INC "+z$
7740 RETURN
7741 LET x$="ADD "+z$+","+z$
7742 RETURN
7743 LET x$="DEC "+z$
7756 RETURN
7757 LET x$="ADD "+z$+",SP"
7796 RETURN
7797 LET x$="POP "+z$
7798 RETURN
7799 LET x$="EX (SP),"+z$
7800 RETURN
7801 LET x$="PUSH "+z$
7804 RETURN
```

Disassembler Z80




```
7805 LET x$="JP ("+z$+")"
7820 RETURN
7821 LET x$="LD SP,"+z$
7827 RETURN
7899 REM *** OP=EDh ***
7900 GO SUB 150
7910 IF op<64 OR op>191 THEN
  GO TO 9000
7920 LET d=0: IF op>=128 THEN
  GO TO 8000
7930 LET k=64: GO SUB 900: IF y
 <>0 THEN  GO TO 7940
7933 IF x=6 THEN  GO TO 9000
7937 LET x$="IN ": LET reg=x:
  GO SUB 1500: LET x$=x$+",(
 C)": GO TO 1000
7940 IF y<>1 THEN  GO TO 7950
7943 IF x=6 THEN  GO TO 9000
7947 LET x$="OUT (C),": LET reg=
 x: GO SUB 1500: GO TO 1000
7950 IF y<>2 THEN  GO TO 7990
7960 LET x$="ADC HL,": IF x/2=
 INT (x/2) THEN  LET x$="SBC
  HL,"
7970 LET x=INT (x/2): LET x$=x$+
 ("BC" AND NOT x)+("DE"
  AND x=1)+("HL" AND x=2)+("
 SP" AND x=3): GO TO 1000
7990 IF y=3 AND (x<>4 AND x
 <>5) THEN LET d=y
8000 LET x$="???": GO SUB 8000+o
 p: GO TO 1000+d
8066 RETURN
8067 LET x$="LD (),BC": RETURN
8068 LET x$="NEG": RETURN
8069 LET x$="RETN": RETURN
8070 LET x$="IM 0": RETURN
8071 LET x$="LD I,A"
8074 RETURN
8075 LET x$="LD BC,()"
8076 RETURN
8077 LET x$="RETI"
8078 RETURN
8079 LET x$="LD R,A"
8082 RETURN
8083 LET x$="LD (),DE"
8085 RETURN
8086 LET x$="IM 1": RETURN
8087 LET x$="LD A,I"
8090 RETURN
8091 LET x$="LD DE,()"
8093 RETURN
8094 LET x$="IM 2": RETURN
8095 LET x$="LD A,R"
8102 RETURN
8103 LET x$="RRD"
8110 RETURN
8111 LET x$="RLD"
8114 RETURN
8115 LET x$="LD (),SP"
8122 RETURN
8123 LET x$="LD SP,()"
8159 RETURN
8160 LET x$="LDI": RETURN
8161 LET x$="CPI": RETURN
8162 LET x$="INI": RETURN
8163 LET x$="OUTI"
8167 RETURN
8168 LET x$="LDD": RETURN
8169 LET x$="CPD": RETURN
8170 LET x$="IND": RETURN
8171 LET x$="OUTD"
8175 RETURN
8176 LET x$="LDIR": RETURN
8177 LET x$="CPIR": RETURN
8178 LET x$="INIR": RETURN
8179 LET x$="OTIR"
8183 RETURN
8184 LET x$="LDDR": RETURN
8185 LET x$="CPDR": RETURN
8186 LET x$="INDR": RETURN
8187 LET x$="OTDR"
8255 RETURN
9000 LET x$="???": GO TO 1000
9900 SAVE "DIZ-80" LINE 1
```

REBIT
COMPUTER
A DIVISION OF G.B.C.

sinclair special

PRESENTA:

GP 50 S STAMPANTE AD IMPATTO

Piccola e compatta, realizzata con standard professionali, ottima per usi obbistici, la stampante GP 50 S è una stampante ad impatto con matrice di stampa 5x8, 35 caratteri/secondo, 32 colonne, incorpora una interfaccia diretta per Sinclair ZX81 e SPECTRUM, densità caratteri 12 CPI, grafica, alimentazione carta a frizione, completa di alimentatore esterno.
Stampa un originale ed una copia.
Set di 96 caratteri ASCII.
Dimensioni: 250x85x215 mm.



GP 50 S STAMPANTE AD IMPATTO

MONITOR A COLORI 10" REBIT

ZX INTERFACE 2

ZX INTERFACE 1

ZX MICRODRIVE

CARTUCCE M/D

SINCLAIR ZX SPECTRUM

ZX MICRODRIVE PER SPECTRUM

Lo ZX Microdrive amplia la possibilità dello ZX Spectrum in quei settori, come quello della didattica e delle piccole applicazioni gestionali, dove è necessaria una veloce ricerca del le informazioni memorizzate su un supporto magnetico.
Ogni cartuccia per Microdrive può contenere da un minimo di 85k byte a 100k byte. Il caricamento di un programma da 48k byte avviene in circa 5 secondi.
Altre caratteristiche sono i comandi di "LOAD, SAVE e VERIFY" per la memorizzazione, il caricamento, e la verifica dei programmi. Il comando "FORMAT" per l'inizializzazione delle cartucce. Il comando "CAT" per ottenere sul video la lista dei files contenuti nella cartuccia, con l'indicazione dello spazio libero disponibile. Comando di "AUTO-RUN" per il caricamento.

N.B.
Per collegare gli ZX Microdrive allo ZX Spectrum è necessario utilizzare l'interfaccia 1.
L'interfaccia può gestire contemporaneamente fino a 8 ZX Microdrive per un totale di 640 k byte.

CARTUCCE M/D SUPPORTI MAGNETICI PER ZX MICRODRIVE

Confezione da 2 e 4 pezzi.

SINCLAIR ZX SPECTRUM

Lo ZX Spectrum, abbatte definitivamente la barriera fra home e personal computer, e riunisce le due tipologie del computer e del videogame. Infatti ad una memoria e ad un sistema operativo da Personal Computer affianca i colori, l'animazione, e il software ricreativo ed educativo tipico dei videogame e degli home computer più evoluti. Il tutto, naturalmente, ad un prezzo significativamente inferiore a quello di una semplice consolle per videogiochi.

Presentare in 3 pagine tutto il "mondo" Sinclair è un'impresa impegnativa: per ogni oggetto illustrato verrebbe voglia di scrivere colonne e colonne di testi esplicativi e applicativi, tante sono le possibilità offerte dal sistema.
Bisogna rinunciare, invece, e attenersi alle caratteristiche fondamentali, alle specifiche tecniche.
È giusto che sia così, anche se "16k, 8 colori, 44 tasti" non dice affatto quanto sia entusiasmante e coinvolgente avere in casa un Sinclair!

ZX — INTERFACE 1

Indispensabile per il collegamento del ZX Microdrive.
Incorpora una interfaccia RS 232 e un sistema di collegamento in rete locale; si connette alla parte posteriore dello ZX Spectrum permettendo comunque il collegamento di altre espansioni periferiche dello ZX Spectrum. La interfaccia seriale RS 232, standard industriale universalmente adottato, permette il collegamento fra lo ZX Spectrum e una ampia gamma di periferiche e di altri computer dotati della medesima interfaccia. Grazie alla RS 232 è anche possibile trasmettere dati sulla linea telefonica utilizzando un modem. Tutte le immagini contenute in uno schermo video possono essere trasferite in circa 3 secondi e il protocollo di collegamento permette a ogni stazione della rete di specificare quali sono le stazioni trasmittenti e riceventi. È inoltre possibile diffondere un messaggio a ogni ZX Spectrum collegato alla rete realizzando un interessante sistema di broadcasting. Ogni Sinclair ZX Spectrum può agire come unità di servizio per altri ZX Spectrum della rete pilotando una stampante ZX o qualsiasi altra periferica collegata tramite la interfaccia RS 232. Ogni ZX Spectrum può inviare e ricevere files dagli altri computer della rete sfruttando al massimo le possibilità offerte dallo ZX Microdrive. La rete può essere costituita da 2 a 64 Spectrum.

ZX — INTERFACE 2

È l'ultima novità in casa Sinclair per lo ZX Spectrum. Permette di utilizzare le nuovissime ZX ROM cartucce software di nuova concezione e di minime dimensioni. È previsto il collegamento per due JOYSTICK di tipo standard "9 poli D".
Con le nuove ZX ROM il programma è immediatamente caricato e pronto all'uso.

MONITOR A COLORI 10" "TUTTO ITALIANO"

Un display ad alta risoluzione professionale dal design molto curato nelle linee e finiture, particolarmente indicato per l'utilizzo con gli home e personal computer per le sue dimensioni ed il prezzo contenuto.
Questo monitor prevede un ingresso con segnale PAL composito più il segnale audio.
Dimensioni: 270x285x370 mm.



GP 500 AS STAMPANTE AD IMPATTO

Dal piacevole design e ad un prezzo incredibilmente contenuto, la GP 500 AS è una stampante ad impatto con matrice di stampa 5x7, 50 caratteri/secondo, 80 colonne, incorpora una interfaccia Seriale RS232C che consente il collegamento diretto alla ZX INTERFACE 1 SINCLAIR dello SPECTRUM.
Caratteri normali ed espansi, grafica, alimentazione della carta a trattori (moduli continui) larghezza 9"3/6. Stampa un originale ed una copia.
Set di 96 caratteri ASCII e 44 caratteri e simboli.
Dimensioni: 315x114x447 mm.

JOYSTICK COMMAND "TRIGA"

Il primo Joystick che ha il pulsante "FIRE" nel punto più naturale per l'uso, difatti il dito indice risponde più prontamente delle altre dita.
Particolarmente adatto per I/F SINCLAIR

"ALLA SCOPERTA DELLO ZX SPECTRUM"

Nato dalla traduzione dei manuali inglesi è costituito da ben 35 capitoli; tratta a fondo tutti i problemi relativi al collegamento ed all'utilizzo dello Spectrum spingendosi fino alla programmazione Basic. La massima chiarezza e la facile consultazione, con l'ausilio della cassetta **DEMO/DIDATTICA** fanno di questo manuale un elemento indispensabile per il possessore dello ZX SPECTRUM.

UN FAVOLOSO LIBRO IN REGALO!

SINCLAIR ZX SPECTRUM

- Grafica a 256x192 punti-schermo.
- 24 linee di 32 caratteri.
- 8 colori indipendenti per testo, sfondo, riquadro.
- Comandi di suono modulabili in frequenza e durata.
- Vera tastiera multifunzione con maiuscole e minuscole. Tutti i tasti con funzione di ripetizione.
- Compatibile con teletext.
- Alta velocità LOAD e SAVE: 16k byte/100 audi.
- Funzioni VERIFY e MERGE per programmi e archivi.
- BASIC Sinclair esteso con funzioni a 1 tasto; controllo di sintassi.
- Ampio software su cassetta.
- 16 k byte ROM

Versione da 16 k RAM e da 48 k RAM.

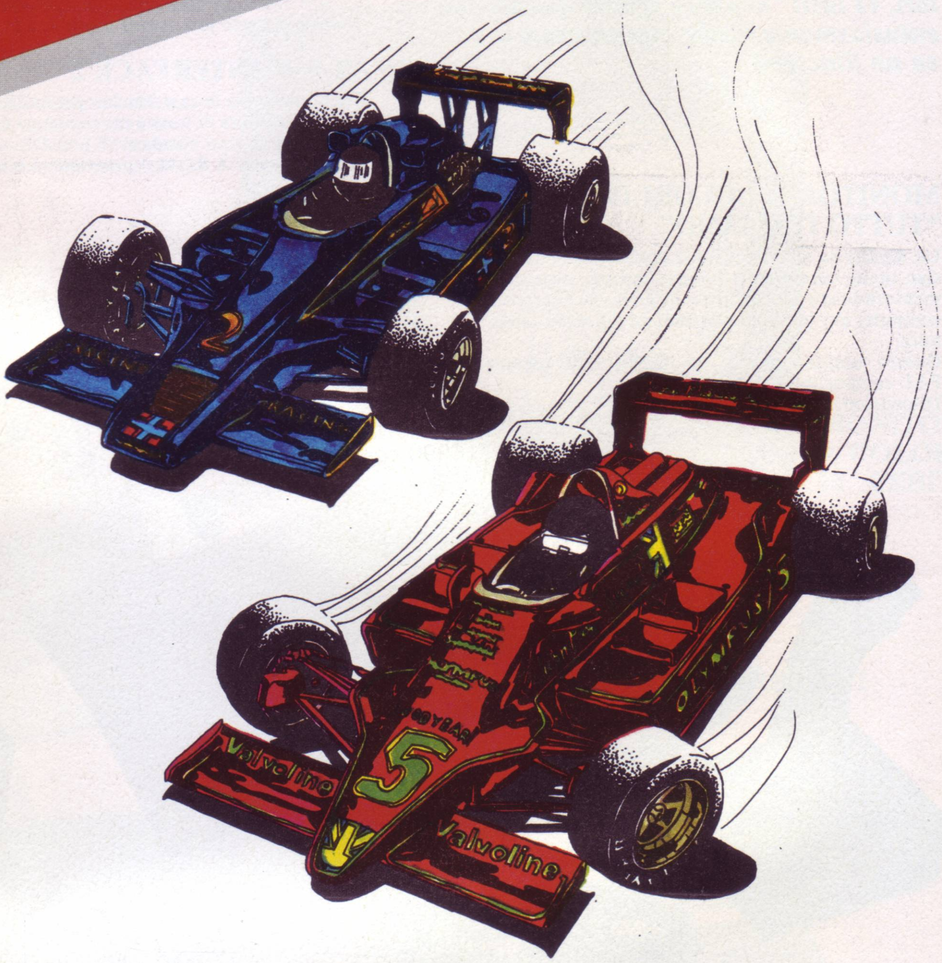
Il tuo Spectrum è preziosissimo: difendilo con la **"SUPER GARANZIA"**! Rebit Computer, distributore per l'Italia dei prodotti **SINCLAIR,** ha messo a punto la nuova **SUPER GARANZIA.**
Acquista lo ZX SPECTRUM presso un Rivenditore Autorizzato e richiedi la **"SUPER GARANZIA"**; oltre ad una perfetta assistenza ed alla certezza del valore del tuo autentico SPECTRUM, avrai dei vantaggi immediati.
Per questo uno SPECTRUM senza la **"SUPER GARANZIA"** è solo un mezzo SPECTRUM!

Formula 1

di **Angelo Motta**

parte prima

Giocate questo "classico" arcade sul vostro ZX81



Formula 1 è un gioco scritto interamente in linguaggio macchina, ad eccezione delle relative istruzioni che ho preferito scrivere in Basic per evitare di aggiungere circa 700 codici macchina ai 1069 già esistenti che i volonterosi si cimenteranno ad inserire nella lunghissima REM iniziale.
Il gioco prende spunto dai diversi arcade di questo tipo esistenti nelle varie sale gioco e consiste nel guidare la propria vettura di "formula 1", situata nel bordo inferiore dello schermo, lungo una pista, sorpassando le auto che il computer fa apparire lungo la strada, senza urtarle ed evitando le uscite di strada. Si hanno a disposizione tre vetture e, dopo altrettanti scontri, il gioco finisce.
Anche se a prima vista il tutto può sembrare molto facile, basta mettersi "al volante" per accorgersi, dopo alcuni istanti, che la velocità dell'auto aumenta sempre più fino ad arrivare a livelli da "suicidi della strada", questo grazie all'utilizzo del linguaggio macchina.
Contrariamente alla maggior parte dei giochi in commercio per lo ZX81, che utilizzano un solo carattere per i singoli soggetti, in formula 1 le auto sono composte da 42 caratteri (matrice da 7 righe x 6 colonne). Questo, se da un lato comporta un notevole livello di grafica che fa dimenticare le limitazioni in tal senso del piccolo ZX81, da un altro fa sì che ben 306 dei codici macchina inseriti nella REM sono relativi ai caratteri grafici delle auto (42 per quella del giocatore e 48 + 48 per quelle del computer), ai caratteri della collisione (126) ed a quelli raffiguranti lo "Start" iniziale (42).
Il programma utilizza tutte le 24 linee dello schermo: le prime 22 per visualizzare la pista dove si svolge il gioco, la 23ª per il punteggio della gara (SCORE) e quello massimo raggiunto (RECORD); la 24ª per il colloquio fra lo ZX81 ed il giocatore prima dell'inizio della gara.
Per avere a disposizione queste ultime due righe si è azzerato il valore della variabile contenente le righe a disposizione dello ZX81 con la riga iniziale:
10 POKE 16418,0
e nel programma si è evitato l'uso della istruzione INPUT sostituendola con la INKEY$ in quanto la prima avrebbe mandato in blocco il programma.
Per quanto riguarda la costruzione del gioco procedere come segue:

— caricare il listato della figura 1 che contiene le istruzioni per la realizzazione della REM iniziale lunga 1073 caratteri contenente il linguaggio macchina (linee 1 — 96) ed il caricatore dello stesso (linee 100 — 140).
— dare il RUN ed inserire i codici del linguaggio macchina riportati nella figura 2. Il programma si fermerà automaticamente dopo aver inserito il 1069° codice.
— sostituire tutte le linee del programma 1, ad eccezione della REM, con quelle riportate nel listato della figura 3; dopodiché il programma è pronto all'uso.

Prima di effettuare una prova, consiglio la registrazione di quanto fatto in modo da proteggersi da eventuali cancellature o blocchi del programma in caso di errore di digitazione dei codici, la cui unica soluzione sarebbe lo spegnimento dello ZX81 con conseguente perdita di tutto il lavoro fatto. Nel caso si verificasse questa eventualità, o un funzionamento errato del programma, per la ricerca degli errori riporto un breve programma, "DUMP" che permette di visualizzare il contenuto delle locazioni di memoria:

```
9900 FOR I = 16514 TO 17580
9910 PRINT I;" = "; PEEK I
9920 NEXT I
```

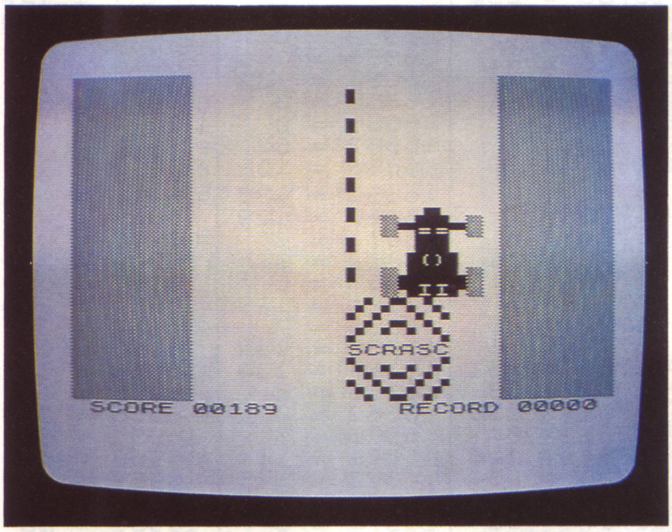
Dopo aver dato il RUN appariranno sullo schermo i contenuti delle prime 22 locazioni di memoria ed il programma si fermerà con il codice errore 5/9910 (non c'è più posto sullo schermo). Dopo aver controllato i codici,

per proseguire il controllo basta premere i tasti CONT e N/L per far ripartire il programma con le successive 22 locazioni di memoria.
La correzione degli errori può essere eseguita in modo diretto con POKE m,n dove m è la locazione di memoria da correggere ed n il codice macchina da inserire.
Per chi fosse interessato al linguaggio macchina e desiderasse studiare il programma, verrà riportato nella seconda parte dell'articolo il relativo Assembler suddiviso in blocchi per renderne più chiara la lettura. Fra questi blocchi merita una particolare attenzione quello relativo allo SCROLL INVERSO (ossia dall'alto verso il basso) che si trova fra le locazioni 17003 e 17030 e che può senz'altro essere utilizzato anche in diversi altri programmi di giochi. Siccome nel programma questo SCROLL è statoi adottato appositamente per il gioco (solo le 18 colonne centrali vengono mosse), per il movimento di tutto lo schermo vengono riportati nelle figure 6 e 7 il programma e l'assembler relativo. La REM iniziale della figura è composta da 41 caratteri. Dopo aver lanciato il programma, apparirà il LIST dello stesso che scrollerà verso il basso.

Possibili modifiche al gioco

Se il gioco vi sembra troppo veloce o le auto a disposizione troppo poche, è possibile modificare questi valori in modo da adattarli alle proprie esigenze personali.
Per quanto riguarda la velocità delle auto, il ciclo relativo all'accelerazione è contenuto fra le locazioni 16984 e 17002. La locazione 16638 contiene il valore (iniziale 127) che caricato nel registro C determina quante volte quel ciclo deve essere eseguito. Ad ogni giro di programma tale valore viene decrementato di 2 (locazioni 16993 e 16994 — DEC A —) sino a raggiungere un valore minimo di 3 che corrisponde alla velocità massima dell'auto (originariamente tale valore scendeva fino a 1, ma la velocità era tale da disturbare notevolmente i riflessi visivi).
Per rallentare la velocità massima dell'auto occorre variare il valore minimo del ciclo contenuto nella locazione 16990 con l'istruzione POKE 16990,n dove n dovrà essere un numero dispari compreso fra 3 (velocità attuale massima) e 127 (velocità iniziale del gioco).
Per quanto riguarda invece il numero della auto a disposizione, tale valore è contenuto nella locazione 16637 ed è generato in fase di partenza fra i byte 16698 e 16702 (LD HL, 16637 — (HL), 3). Il valore del byte 16637 viene inoltre controllato fra le locazioni 16781 e 16788 e, se l'auto del giocatore è la prima, viene stampato lo START sulla pista. Pertanto, per aumentare le auto a disposizione dovranno essere variate 2 locazioni, la 16702 (generazione del n.ro auto) e la 16786 (controllo se è la prima auto), entrambe con l'istruzione diretta POKE 16702,n e POKE 16786,n dove n sarà il nuovo numero delle auto (teoricamente è possibile inserire fino a 255 auto).
Un'ultima avvertenza: nel caso il programma venga registrato dopo averlo fatto girare, portare a 0 i byte 16635 e 16636 (POKE m,0) che contengono il valore del record raggiunto in quanto il programma non prevede l'azzeramento automatico degli stessi in fase di partenza.

Formula 1
versione per ZX81 16K
caricatore linguaggio macchina




```
   1 REM 1234567890123456789012З
4567890123456789О1234
   2 REM 12345678901234567890123
4567890123456789О1234
   3 REM 12345678901234567890123
45678901234567890123
   4 REM 12345678901234567890123
45678901234567890123 4
   5 REM 12345678901234567890123
456789012345678901234
   6 REM 12345678901234567890123
456789012345678901234
   7 REM 12345678901234567890123
456789012345678901234
   8 REM 12345678901234567890123
456789012345678901234
   9 REM 12345678901234567890123
456789012345678901234
  10 REM 12345678901234567890123
456789012345678901234
  11 REM 12345678901234567890123
456789012345678901234
  12 REM 12345678901234567890123
456789012345678901234
  13 REM 12345678901234567890123
456789012345678901234
  14 REM 12345678901234567890123
456789012345678901234
  15 REM 12345678901234567890123
456789012345678901234
  16 REM 12345678901234567890123
456789012345678901234
  17 REM 12345678901234567890123
456789012345678901234
  18 REM 12345678901234567890123
456789012345678901234
  19 REM 12345678901234567890123
456789012345678901234
  90 FOR I=16514 TO 17578
  92 IF PEEK I=118 THEN POKE I,0
  93 NEXT I
  94 POKE 16511,0045
  96 POKE 16512,0004
 100 FOR I=16514 TO 17580
 110 INPUT A
 120 IF PEEK 16418=2 THEN SCROLL

 130 PRINT I;" = ";A
 140 POKE I,A
 150 NEXT I
 160 STOP
1000 REM QUESTO PROGRAMMA CONSEN
TE DI CREARE UNA REM INIZIALE DI
1069 CARATTERI.
1010 REM DARE IL RUN IN FAST E
DOPO CIRCA MEZZO MINUTO LA REM
SARA" CREATA.
1020 REM INSERIRE QUINDI I 1069
CODICI DEL LINGUAGGIO MACCHINA E
IL PROGRAMMA SI FERMERA" CON
9/160.
1030 REM CANCELLARE LE LINEE DA
100 A 1030 ED INSERIRE IL RESTO
DEL PROGRAMMA BASIC.
```

LOAD

Formula 1

Formula 1 versione ZX81 16K codici da inserire nella REM di linea 1

```
16514 = 249
16515 = 1
16516 = 135
16517 = 4
16518 = 128
16519 = 128
16520 = 135
16521 = 4
16522 = 133
16523 = 7
16524 = 128
16525 = 128
16526 = 132
16527 = 5
16528 = 0
16529 = 135
16530 = 128
16531 = 128
16532 = 4
16533 = 0
16534 = 0
16535 = 133
16536 = 144
16537 = 145
16538 = 5
16539 = 0
16540 = 128
16541 = 129
16542 = 7
16543 = 132
16544 = 130
16545 = 128
16546 = 128
16547 = 132
16548 = 148
16549 = 148
16550 = 7
16551 = 128
16552 = 0
16553 = 2
16554 = 7
16555 = 132
16556 = 1
16557 = 0
16558 = 5
16559 = 7
16560 = 1
16561 = 2
16562 = 7
16563 = 0
16564 = 7
16565 = 5
16566 = 133
16567 = 3
16568 = 5
16569 = 2
16570 = 7
16571 = 133
16572 = 5
16573 = 3
16574 = 5
16575 = 0
16576 = 5
16577 = 0
16578 = 7
16579 = 5
16580 = 133
16581 = 132
16582 = 1
16583 = 0
16584 = 5
16585 = 133
16586 = 5
16587 = 3
16588 = 1
16589 = 0
16590 = 1
16591 = 0
16592 = 1
16593 = 1
16594 = 2
16595 = 0
16596 = 1
16597 = 0
16598 = 1
16599 = 133
16600 = 42
16601 = 12
16602 = 64
16603 = 237
16604 = 75
16605 = 130
16606 = 64
16607 = 9
16608 = 17
16609 = 132
16610 = 64
16611 = 6
16612 = 7
16613 = 197
16614 = 235
16615 = 1
16616 = 6
16617 = 0
16618 = 237
16619 = 176
16620 = 1
16621 = 27
16622 = 0
16623 = 235
16624 = 9
16625 = 193
16626 = 16
16627 = 241
16628 = 201
16629 = 90
16630 = 74
16631 = 71
16632 = 74
16633 = 0
16634 = 0
16635 = 47
16636 = 2
16637 = 3
16638 = 127
16639 = 13
16640 = 56
16641 = 40
16642 = 52
16643 = 55
16644 = 42
16645 = 0
16646 = 28
16647 = 28
16648 = 28
16649 = 28
16650 = 28
16651 = 55
16652 = 42
16653 = 40
16654 = 52
16655 = 55
16656 = 41
16657 = 0
16658 = 28
16659 = 28
16660 = 28
16661 = 28
16662 = 28
16663 = 42
16664 = 12
16665 = 64
16666 = 1
16667 = 216
16668 = 2
16669 = 9
16670 = 17
16671 = 0
16672 = 65
16673 = 235
16674 = 1
16675 = 11
16676 = 0
16677 = 237
16678 = 176
16679 = 235
16680 = 43
16681 = 34
16682 = 247
16683 = 64
16684 = 1
16685 = 8
16686 = 0
16687 = 9
16688 = 14
16689 = 12
16690 = 235
16691 = 237
16692 = 176
16693 = 27
16694 = 237
16695 = 83
16696 = 245
16697 = 64
16698 = 33
16699 = 253
16700 = 64
16701 = 54
16702 = 3
16703 = 42
16704 = 12
16705 = 64
16706 = 1
16707 = 7
16708 = 0
16709 = 9
16710 = 17
16711 = 15
16712 = 0
16713 = 14
16714 = 22
16715 = 6
16716 = 18
16717 = 35
16718 = 54
16719 = 0
16720 = 16
16721 = 251
16722 = 25
16723 = 13
16724 = 32
16725 = 245
16726 = 33
16727 = 254
16728 = 64
16729 = 54
16730 = 127
16731 = 35
16732 = 54
16733 = 13
16734 = 42
16735 = 12
16736 = 64
16737 = 17
16738 = 26
16739 = 0
16740 = 6
16741 = 2
16742 = 197
16743 = 14
16744 = 22
16745 = 6
16746 = 7
16747 = 35
16748 = 54
16749 = 8
16750 = 16
16751 = 251
16752 = 25
16753 = 13
16754 = 32
16755 = 245
16756 = 1
16757 = 169
16758 = 2
16759 = 237
16760 = 66
16761 = 193
16762 = 16
16763 = 234
16764 = 42
16765 = 12
16766 = 64
16767 = 17
16768 = 66
16769 = 0
16770 = 1
16771 = 17
16772 = 0
16773 = 9
16774 = 6
16775 = 11
16776 = 54
16777 = 5
16778 = 25
16779 = 16
16780 = 251
16781 = 33
16782 = 253
16783 = 64
16784 = 126
16785 = 254
16786 = 3
16787 = 32
16788 = 50
16789 = 42
16790 = 12
16791 = 64
16792 = 1
16793 = 208
16794 = 0
16795 = 9
16796 = 54
16797 = 7
16798 = 6
16799 = 12
16800 = 35
16801 = 54
16802 = 3
16803 = 16
16804 = 251
16805 = 35
16806 = 54
16807 = 132
16808 = 1
16809 = 20
16810 = 0
16811 = 9
16812 = 17
16813 = 174
16814 = 64
16815 = 6
16816 = 3
16817 = 197
16818 = 235
16819 = 1
16820 = 14
16821 = 0
16822 = 237
16823 = 176
16824 = 1
16825 = 19
16826 = 0
16827 = 235
16828 = 9
16829 = 193
16830 = 16
16831 = 241
16832 = 6
16833 = 14
16834 = 54
16835 = 3
16836 = 35
16837 = 16
16838 = 251
16839 = 33
16840 = 130
16841 = 64
16842 = 54
16843 = 249
16844 = 35
16845 = 54
16846 = 1
16847 = 205
16848 = 216
16849 = 64
16850 = 201
16851 = 0
16852 = 0
16853 = 7
16854 = 132
16855 = 0
16856 = 0
16857 = 8
16858 = 132
16859 = 130
16860 = 129
16861 = 7
16862 = 8
16863 = 8
16864 = 129
16865 = 7
16866 = 132
16867 = 130
16868 = 8
16869 = 0
16870 = 0
16871 = 144
16872 = 145
16873 = 0
16874 = 0
16875 = 10
16876 = 0
16877 = 128
16878 = 128
16879 = 0
16880 = 10
16881 = 8
16882 = 131
16883 = 148
16884 = 148
16885 = 131
16886 = 8
16887 = 0
16888 = 0
16889 = 129
16890 = 130
16891 = 0
16892 = 0
16893 = 0
16894 = 0
16895 = 0
16896 = 0
16897 = 0
16898 = 0
16899 = 0
16900 = 0
16901 = 3
16902 = 3
16903 = 0
16904 = 0
16905 = 8
16906 = 132
16907 = 174
16908 = 174
16909 = 7
16910 = 8
16911 = 8
16912 = 129
16913 = 128
16914 = 128
16915 = 130
16916 = 0
16917 = 0
16918 = 133
16919 = 144
16920 = 145
16921 = 5
16922 = 0
16923 = 0
16924 = 0
16925 = 128
16926 = 128
16927 = 0
16928 = 0
16929 = 8
16930 = 3
16931 = 148
16932 = 148
16933 = 3
16934 = 8
16935 = 9
16936 = 0
16937 = 129
16938 = 130
16939 = 0
16940 = 9
16941 = 0
16942 = 0
16943 = 0
16944 = 0
16945 = 0
16946 = 0
16947 = 5
16948 = 0
16949 = 3
16950 = 66
16951 = 206
16952 = 0
16953 = 0
16954 = 0
16955 = 42
16956 = 247
16957 = 64
16958 = 6
16959 = 5
16960 = 54
16961 = 28
16962 = 43
16963 = 16
16964 = 251
16965 = 58
16966 = 52
16967 = 64
16968 = 50
16969 = 55
16970 = 66
16971 = 33
16972 = 56
16973 = 66
16974 = 54
16975 = 0
16976 = 33
16977 = 51
16978 = 66
16979 = 54
16980 = 20
16981 = 35
```

Address		Value
16982	=	54
16983	=	0
16984	=	33
16985	=	254
16986	=	64
16987	=	78
16988	=	121
16989	=	254
16990	=	3
16991	=	40
16992	=	3
16993	=	61
16994	=	61
16995	=	119
16996	=	6
16997	=	255
16998	=	16
16999	=	254
17000	=	13
17001	=	32
17002	=	249
17003	=	42
17004	=	12
17005	=	64
17006	=	1
17007	=	156
17008	=	2
17009	=	9
17010	=	6
17011	=	21
17012	=	197
17013	=	17
17014	=	33
17015	=	0
17016	=	235
17017	=	25
17018	=	235
17019	=	1
17020	=	18
17021	=	0
17022	=	237
17023	=	176
17024	=	17
17025	=	51
17026	=	0
17027	=	237
17028	=	82
17029	=	193
17030	=	16
17031	=	236
17032	=	1
17033	=	42
17034	=	0
17035	=	9
17036	=	126
17037	=	254
17038	=	0
17039	=	32
17040	=	4
17041	=	54
17042	=	5
17043	=	24
17044	=	2
17045	=	54
17046	=	0
17047	=	205
17048	=	187
17049	=	2
17050	=	124
17051	=	198
17052	=	2
17053	=	56
17054	=	47
17055	=	68
17056	=	77
17057	=	205
17058	=	189
17059	=	7
17060	=	78
17061	=	42
17062	=	130
17063	=	64
17064	=	84
17065	=	93
17066	=	121
17067	=	254
17068	=	33
17069	=	32
17070	=	9
17071	=	43
17072	=	19
17073	=	19
17074	=	19
17075	=	19
17076	=	19
17077	=	0
17078	=	24
17079	=	5
17080	=	254
17081	=	36
17082	=	32
17083	=	18
17084	=	35
17085	=	34
17086	=	130
17087	=	64
17088	=	42
17089	=	12
17090	=	64
17091	=	25
17092	=	17
17093	=	33
17094	=	0
17095	=	6
17096	=	7
17097	=	54
17098	=	0
17099	=	25
17100	=	16
17101	=	251
17102	=	33
17103	=	255
17104	=	64
17105	=	126
17106	=	254
17107	=	0
17108	=	40
17109	=	4
17110	=	61
17111	=	119
17112	=	24
17113	=	33
17114	=	42
17115	=	12
17116	=	64
17117	=	237
17118	=	91
17119	=	130
17120	=	64
17121	=	25
17122	=	126
17123	=	254
17124	=	0
17125	=	40
17126	=	5
17127	=	254
17128	=	5
17129	=	194
17130	=	11
17131	=	68
17132	=	35
17133	=	35
17134	=	35
17135	=	35
17136	=	35
17137	=	126
17138	=	254
17139	=	0
17140	=	40
17141	=	5
17142	=	254
17143	=	5
17144	=	194
17145	=	11
17146	=	68
17147	=	205
17148	=	216
17149	=	64
17150	=	0
17151	=	58
17152	=	52
17153	=	66
17154	=	254
17155	=	0
17156	=	40
17157	=	19
17158	=	79
17159	=	6
17160	=	0
17161	=	42
17162	=	12
17163	=	64
17164	=	9
17165	=	237
17166	=	91
17167	=	53
17168	=	66
17169	=	235
17170	=	14
17171	=	6
17172	=	237
17173	=	176
17174	=	34
17175	=	53
17176	=	66
17177	=	33
17178	=	51
17179	=	66
17180	=	126
17181	=	61
17182	=	119
17183	=	254
17184	=	0
17185	=	32
17186	=	81
17187	=	35
17188	=	126
17189	=	254
17190	=	0
17191	=	32
17192	=	51
17193	=	42
17194	=	55
17195	=	66
17196	=	78
17197	=	35
17198	=	34
17199	=	55
17200	=	64
17201	=	121
17202	=	254
17203	=	127
17204	=	56
17205	=	5
17206	=	33
17207	=	211
17208	=	65
17209	=	24
17210	=	3
17211	=	33
17212	=	3
17213	=	66
17214	=	34
17215	=	53
17216	=	66
17217	=	42
17218	=	55
17219	=	66
17220	=	78
17221	=	35
17222	=	34
17223	=	55
17224	=	66
17225	=	33
17226	=	52
17227	=	66
17228	=	121
17229	=	254
17230	=	127
17231	=	56
17232	=	4
17233	=	54
17234	=	10
17235	=	24
17236	=	2
17237	=	54
17238	=	19
17239	=	43
17240	=	54
17241	=	8
17242	=	24
17243	=	24
17244	=	42
17245	=	55
17246	=	66
17247	=	78
17248	=	35
17249	=	34
17250	=	55
17251	=	66
17252	=	121
17253	=	214
17254	=	22
17255	=	56
17256	=	2
17257	=	24
17258	=	250
17259	=	198
17260	=	27
17261	=	33
17262	=	51
17263	=	66
17264	=	119
17265	=	35
17266	=	54
17267	=	0
17268	=	42
17269	=	247
17270	=	64
17271	=	126
17272	=	254
17273	=	37
17274	=	32
17275	=	5
17276	=	54
17277	=	28
17278	=	43
17279	=	24
17280	=	246
17281	=	60
17282	=	119
17283	=	42
17284	=	249
17285	=	64
17286	=	35
17287	=	34
17288	=	249
17289	=	64
17290	=	195
17291	=	68
17292	=	66
17293	=	135
17294	=	131
17295	=	131
17296	=	131
17297	=	131
17298	=	4
17299	=	133
17300	=	135
17301	=	131
17302	=	131
17303	=	4
17304	=	5
17305	=	133
17306	=	133
17307	=	135
17308	=	4
17309	=	5
17310	=	5
17311	=	133
17312	=	133
17313	=	2
17314	=	1
17315	=	5
17316	=	5
17317	=	133
17318	=	2
17319	=	3
17320	=	3
17321	=	1
17322	=	5
17323	=	2
17324	=	3
17325	=	3
17326	=	3
17327	=	3
17328	=	1
17329	=	0
17330	=	0
17331	=	0
17332	=	0
17333	=	0
17334	=	0
17335	=	7
17336	=	3
17337	=	3
17338	=	3
17339	=	3
17340	=	132
17341	=	5
17342	=	7
17343	=	3
17344	=	3
17345	=	132
17346	=	133
17347	=	5
17348	=	5
17349	=	7
17350	=	132
17351	=	133
17352	=	133
17353	=	5
17354	=	5
17355	=	130
17356	=	129
17357	=	133
17358	=	133
17359	=	5
17360	=	130
17361	=	131
17362	=	131
17363	=	129
17364	=	133
17365	=	130
17366	=	131
17367	=	131
17368	=	131
17369	=	131
17370	=	129
17371	=	0
17372	=	0
17373	=	0
17374	=	0
17375	=	0
17376	=	0
17377	=	135
17378	=	1
17379	=	6
17380	=	134
17381	=	2
17382	=	4
17383	=	1
17384	=	6
17385	=	135
17386	=	4
17387	=	134
17388	=	2
17389	=	6
17390	=	0
17391	=	1
17392	=	2
17393	=	0
17394	=	134
17395	=	56
17396	=	40
17397	=	55
17398	=	38
17399	=	56
17400	=	40
17401	=	134
17402	=	0
17403	=	4
17404	=	135
17405	=	0
17406	=	6
17407	=	4
17408	=	134
17409	=	2
17410	=	1
17411	=	6
17412	=	135
17413	=	2
17414	=	4
17415	=	134
17416	=	6
17417	=	135
17418	=	1
17419	=	6
17420	=	10
17421	=	197
17422	=	17
17423	=	141
17424	=	67
17425	=	6
17426	=	2
17427	=	197
17428	=	42
17429	=	12
17430	=	64
17431	=	237
17432	=	75
17433	=	130
17434	=	64
17435	=	9
17436	=	205
17437	=	227
17438	=	64
17439	=	14
17440	=	40
17441	=	6
17442	=	255
17443	=	16
17444	=	254
17445	=	13
17446	=	32
17447	=	249
17448	=	193
17449	=	16
17450	=	232
17451	=	193
17452	=	16
17453	=	223
17454	=	42
17455	=	12
17456	=	64
17457	=	237
17458	=	75
17459	=	130
17460	=	64
17461	=	9
17462	=	205
17463	=	227
17464	=	64
17465	=	58
17466	=	253
17467	=	64
17468	=	61
17469	=	50
17470	=	253
17471	=	64
17472	=	254
17473	=	0
17474	=	40
17475	=	15
17476	=	14
17477	=	255
17478	=	6
17479	=	255
17480	=	16
17481	=	254
17482	=	13
17483	=	32
17484	=	249
17485	=	205
17486	=	63
17487	=	65
17488	=	195
17489	=	79
17490	=	66
17491	=	42
17492	=	251
17493	=	64
17494	=	237
17495	=	91
17496	=	249
17497	=	64
17498	=	237
17499	=	82
17500	=	48
17501	=	53
17502	=	6
17503	=	30

LOAD

Formula 1

```
17504 = 197   17511 = 9    17518 = 251
17505 = 42    17512 = 6    17519 = 205
17506 = 12    17513 = 6    17520 = 165
17507 = 64    17514 = 54   17521 = 68
17508 = 1     17515 = 0    17522 = 17
17509 = 234   17516 = 35   17523 = 17
17510 = 2     17517 = 16   17524 = 65

17525 = 235   17544 = 247  17563 = 205
17526 = 1     17545 = 64   17564 = 165
17527 = 7     17546 = 237  17565 = 68
17528 = 0     17547 = 91   17566 = 37
17529 = 237   17548 = 245  17567 = 32
17530 = 184   17549 = 64   17568 = 250
17531 = 205   17550 = 1    17569 = 205
17532 = 165   17551 = 5    17570 = 58
17533 = 68    17552 = 0    17571 = 65
17534 = 193   17553 = 237  17572 = 201
17535 = 16    17554 = 184  17573 = 14
17536 = 223   17555 = 33   17574 = 16
17537 = 42    17556 = 249  17575 = 16
17538 = 249   17557 = 64   17576 = 254
17539 = 64    17558 = 54   17577 = 13
17540 = 34    17559 = 0    17578 = 32
17541 = 251   17560 = 35   17579 = 251
17542 = 64    17561 = 54   17580 = 201
17543 = 42    17562 = 0
```

Formula 1 versione per ZX81 16K programma principale

```
   1 REM RAND ▀ █ ▀▀ █ ▄ █▀▀█
  ()      ==
▌ ▘ ▌E£RND GOSUB ?▙RND▖) ▘RND ▟VAL
FOR ▝ ▗  GOSUB ▛.  FOR ▖AT ( LET
 TAN ????     ▀▛$SCORE 00000RECOR
D 00000E£RND▘ ** ▝▖) INKEY$ FOR ▝ "
 GOSUB ▛ FOR F6 RUN RND▝▒ ▖:£ F
OR GOSUB ▛. GOSUB ? PRINT RND5 C
LEAR RNDQ E£RND▘▛ ▖)? : -▞>7Q ((C
LS ;$4 PRINT 5 RETURN RNDQ▛7Q$E£
RND) ▗ ▘VAL : -▞7Q▒( CLS ;$4 PRI
NT ▘▛ ▗ GOSUB PIAT ( REM E£RND)PI
 ) ▖"Q ;( CLS 5 CLEAR RND£RND▘
SQR ▖Q▛▗£7Q▘( CLS 7Q▛ = ▖)▛RND▗
▀VAL FOR ▘:Q▘7( GOSUB ▛ < FOR ▖AT
( LET ▘:Q▘7( CLS 5▙RNDQ RAND 7Q▘
LN **RNDTAN
()      ==      ()      ==
  II
       NPI?   E RUN RND▟
Q0F( CLS UORNDMRPI5SPIQ 5NPIQ=7Q
 5 RETURN RND?? RETURN ▀C▀XX?▘ C
OPY ( RETURN $4 RAND E£RND▘▛ ▖+
VAL )5 FOR ; FOR ▝ > GOSUB ▛)N
 GOSUB ?AT ( GOTO ▘E ▟ / ▘Q LN ▛▘
?LEN ▝SJ??LN ▛▘?E▙RND??? RETURN
54▖F<<<<< /▌ RETURN 84>76▙RNDE£R
ND;)5 ▗▛Q ;( CLS 5 COPY RND?/5E£
RND GOSUB ?▙RND;▌TAB "?????▌TAB
 ?LN **RND UOPI RETURN  C<?▘ E£
RND▖ GOSUB ?PPI FOR :▞ GOSUB ▛6P
PI5NPI??7RPI?76RRND? RETURN ▛S 5P
EEK INKEY$/▀5▀PI6PPIERPI?76RPI5Q
PI? RETURN ▛S, Q▀/ ▘Q<FQ▒//ERPI?76
RPI?CHR$ -5 ▘/IF LEN .5NPI??Q E
RUN RNDQF/ PLOT W?E RAND RND76 R
AND RND??PI
              SCRASC
▘ ▞ ▗ ▘ ▖▘VAL )▛?▘▘VAL E£RND GO
SUB ?▙RND▖LN STOP RND:C▘ COPY (
RETURN $4 RAND AT ( CONT AT ( TO
 E£RND GOSUB ?▙RND▖LN STOP RNDU
CLEAR RNDXM CLEAR RND-RETURN  C?
: COPY ( COPY ( RETURN $4 RAND L
N ZINKEY$??PIE CLS RND GOSUB ? R
AND RND GOSUB ?KP▘2VAL E£RND▘ RE
M ▝▖▘Q ?( CLS LN ▛?))INKEY$ FOR
 ▘▛  GOSUB ▛LN ▛?AT ( TO E RAND
RND6 CLS RNDE RUN RND GOSUB ? PR
INT RND▘▌ GOSUB ▛5 RAND RNDQ 7Q
 LN ▛?94 IF LN UINKEY$TAN :(( RE
TURN $4 CLS TAN

  10 POKE 16418,0
  20 PRINT TAB 12;"██████████";TA
B 12;"FORMULA 1";TAB 12;"██████
██"
  30 PRINT ,,"HAI A DISPOSIZIONE
 TRE AUTO PER PARTECIPARE AL GRA
N PREMIO."
  40 PRINT "DEVI SORPASSARE LE A
UTO CHE IN- CONTRERAI LUNGO LA P
ISTA SENZA URTARLE E SENZA USCI
RE DI STRA- DA."
  50 PRINT "IL GIOCO TERMINA QUA
NDO AVRAI DISTRUTTO LE TUE AUT
O."
  60 PRINT "COMANDI:"
  70 PRINT "5) SPOSTI L'AUTO A S
INISTRA"
  80 PRINT "8) SPOSTI L'AUTO A D
ESTRA"
  90 PRINT ,,"BUON DIVERTIMENTO"
 100 PRINT ,,,,"PREMI UN TASTO"
 110 IF INKEY$="" THEN GOTO 110
 120 LET L=USR 16663
 130 PRINT AT 23,0;"PREMI UN TAS
TO PER PARTIRE"
 140 IF INKEY$="" THEN GOTO 140
 150 PRINT AT 23,0;"
                 "
 160 LET L=USR 16955
 170 GOTO 130
 190 SAVE "FORMULA 1"
 200 RUN
```

segue

Nel prossimo numero di SUPERSINC il listato Assembler di Formula 1

Soccorso aereo

di **C. Wakelin** e **T. Parklett**
trad. e adatt.
di **Paolo Maffei**



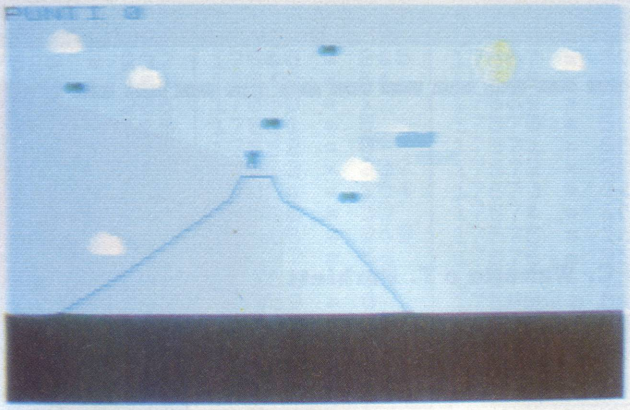
Un uomo da salvare in cima ad un'alta montagna...

Un gioco dai comandi estremamente semplici, in cui il vostro compito consiste nel recupero di un omino rimasto isolato in cima ad una montagna. Col tasto W potrete spostarvi verso l'alto, mentre con S la vostra quota diminuisce. Perché il salvataggio riesca dovete passare appena poco più in alto dell'omino; dopo di che passeete in un altro schermo dove, tenendo premuto il tasto B, atterrerete mettendo in salvo il vostro passeggero.
Guadagnerete punti anche passando, quando possibile, sopra i serbatoi di carburante che galleggiano in aria. Quanto agli ostacoli, vedete di evitare le nuvole, imprevedibilmente consistenti...
La fine del gioco è piuttosto atipica... non c'è. Potete difatti continuare a giocare fino a quando avete voglia, ma attenti! Al più piccolo errore perderete tutti i punti faticosamente conquistati.

Soccorso aereo versione per ZX Spectrum 16K/48K

```
   1 GO SUB 9000
   2 PAPER 5: CLS : PLOT  INK 3;
     0,40: DRAW  INK 3;255,0
   3 LET hs=0
   4 LET sc=0
   6 LET vi=2
   7 LET mo=0
   8 LET uo=0
  10 PLOT 20,40: DRAW 30,20:
     DRAW 40,30: DRAW 5,10:
     DRAW 12,0: DRAW 5,-10:
     DRAW 24,-15: DRAW 25,-30:
     DRAW 3,-5
 100 PRINT AT 17,0;"{32SG8}"'"
     {32SG8}"'"{32SG8}"'"
     {32SG8}"'"{32SG8}"
 150 PRINT AT 3,6; INK 7;"AB";
     AT 4,6; INK 7;"CD"
 160 PRINT AT 8,17; INK 7;"AB";
     AT 9,17; INK 7;"CD"
 170 PRINT AT 6,19; INK 7;"AB";
     AT 7,19; INK 7;"CD"
 180 PRINT AT 2,28; INK 7;"AB";
     AT 3,28; INK 7;"CD"
 190 PRINT AT 1,2; INK 7;"AB";
     AT 2,2; INK 7;"CD"
 200 PRINT AT 12,4; INK 7;"AB";
     AT 13,4; INK 7;"CD"
 600 FOR f=1 TO 9 STEP 2:
     CIRCLE  INK 6;200,150,f:
     NEXT f
 680 PRINT AT 8,12;"I"
```

```
685 PRINT AT 6,13;"J";AT 2,16;"
    J"
690 PRINT AT 4,3;"J";AT 10,17;"
    J"
800 LET x=28: LET y=14
810 PRINT AT y,x;" EF ";AT y-1,
    x;"    "
815 PRINT AT y+1,x;"    "
816 PRINT AT 0,0;"PUNTI ";sc
820 LET x=x-1: BEEP .01,-5
822 IF INKEY$="w" THEN  LET y=y
    -1
823 IF y<1 THEN  LET mo=1
824 IF INKEY$="s" THEN  LET y=y
    +1: IF y>15 THEN  LET mo=1
825 IF x<1 THEN  GO TO 3000
830 IF y=3 AND x=6 OR y=4 AND x
    =6 THEN  LET mo=1
840 IF y=6 AND x=19 OR y=7
     AND x=19 THEN  LET mo=1
850 IF y=8 AND x=17 OR y=9
     AND x=17 THEN  LET mo=1
860 IF y=2 AND x=28 OR y=3
     AND x=28 THEN  LET mo=1
870 IF y=1 AND x=2 OR y=2 AND x
    =2 THEN  LET mo=1
880 IF y=12 AND x=4 OR y=13
     AND x=4 THEN  LET mo=1
890 IF y=7 AND x=12 THEN  LET s
    c=sc+10: LET uo=1
910 IF y=2 AND x=16 OR y=6
     AND x=13 OR y=4 AND x=3
     OR y=10 AND x=17 THEN
     LET sc=sc+30
1000 IF mo=1 THEN  PRINT AT y,x;
    "GH  ": LET vi=vi-1
1010 IF vi=0 THEN  GO TO 2000
1500 GO TO 810
2000 BEEP 1,3: BEEP .7,4: BEEP .
    2,5: BEEP .3,4: BEEP .6,4:
     BEEP .5,3: BEEP 1,3
2480 FOR f=1 TO 100: NEXT f
2490 GO TO 2
3000 CLS : PLOT 0,40: DRAW
     INK 4;255,0
3005 LET x=29
3010 PRINT AT 16,0;"{32SG8}"'"
    {32SG8}"'"{32SG8}"'"
    {32SG8}"'"{32SG8}"'"
    {32SG8}"
3020 IF uo=0 THEN  PRINT AT 0,0;
    "SALVATAGGIO NON RIUSCITO,R
    IPROVA": PAUSE 100: CLS :
     GO TO 7
```




```
3030 IF INKEY$="b" THEN  LET y=y
    +1
3035 PRINT AT y,x;" EF ";AT y-1,
    x;"    "
3040 IF x<2 THEN  LET mo=1
3045 LET x=x-1: BEEP .01,0:
     BEEP .01,-10
3050 IF  y>14 THEN  GO TO 4000
3060 IF mo=1 THEN  PRINT AT y,x;
    " GH ": GO TO 2000
3500 GO TO 3030
4000 LET sc=sc+100: PRINT
     FLASH 1;AT 12,9;"BRAVO":
     FOR f=1 TO 200: NEXT f:
     CLS : GO TO 6
9000 FOR c=1 TO 10: FOR n=0 TO 7
    : READ d: POKE USR (CHR$ (1
    43+c))+n,d: NEXT n: NEXT c
9010 DATA 0,0,0,7,15,31,63,63
9020 DATA 0,0,0,128,192,248,252,
    252
9030 DATA 63,127,127,63,31,0,0,0
9040 DATA 252,254,254,254,252,0,
    0,0
9050 DATA 63,85,255,255,255,255,
    127,0
9060 DATA 135,255,255,255,255,25
    4,252,0
9070 DATA 2,61,203,212,35,44,20,
    3
9080 DATA 250,22,34,162,92,144,4
    8,192
9090 DATA 90,90,126,60,60,60,36,
    36
9100 DATA 0,0,102,255,255,255,10
    2,0
9110 RETURN
```

Poker

di **Alberto Chiericati**

Il brivido dell'azzardo, senza il rischio della bancarotta!



Il seguente programma consente di giocare a poker tra un giocatore ed il computer.

Le carte vengono scelte da un mazzo di 24, partenti dal 9 per arrivare fino all'asso.

Durante la prima parte del gioco, lo Spectrum distribuisce le carte a sè ed al giocatore, ed a seconda della combinazione di cui dispone decide e cambia quelle inutili. Successivamente tocca al giocatore, dopo che sono state visualizzate le carte, decidere quali scartare (il massino è di 4).

Dopo la seconda e definitiva visualizzazione del gioco, l'operatore è chiamato a fare la sua puntata, che può andare da 100 dollari a 3000 dollari.

A questo punto il Sinclair agisce nei seguenti modi:

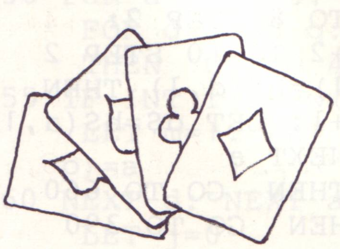
VEDE quando dispone di almeno una coppia e la puntata non è molto alta; altrimenti vede ugualmente se però ha almeno un tris.

PASSA quando la puntata del giocatore è troppo alta e non dispone come minimo di un tris.

RILANCIA DI 1000$: 1) Nel caso non abbia nessuna combinazione valida (quindi cerca di bluffare).

2) Se ha più di un tris.

Dopo il successivo rilancio di 1000 dollari da parte del computer, l'operatore deve decidere se vedere (digitando VEDO), oppure se passare (digitando PASSO). L'ordine con il quale lo Spectrum assegna le combinazioni valide è il seguente: coppia, doppia coppia, tris, scala, full, colore, poker, scala reale.



Linee	Commento
15-25	Generazione caratteri grafici
45-90	Distribuzione carte
100-190	Verifica combinazioni carte dello Spectrum
310-370	Verifica della scala colore
390-440	Verifica scala
445-540	Nessuna combinazione valida (tiene carta più alta e scarta le altre)
560-590	Verifica coppa e tris
610-660	Verifica doppia coppia
670-760	Cambio carte giocatore
770-788	Puntata del giocatore
790-856	Verifica configurazioni carte Spectrum e giocatore
860-900	Verifica scala
905-930	Verifica scala colore
940	Verifica scala reale
950-1050	Assegnazione punteggi
1055-1090	Decisione dello Spectrum se vedere o passare
1100-1146	Determinazione vincitore
2001-3040	Routine per assegnazione e cambio carte
4000-5840	Generazione grafica gioco

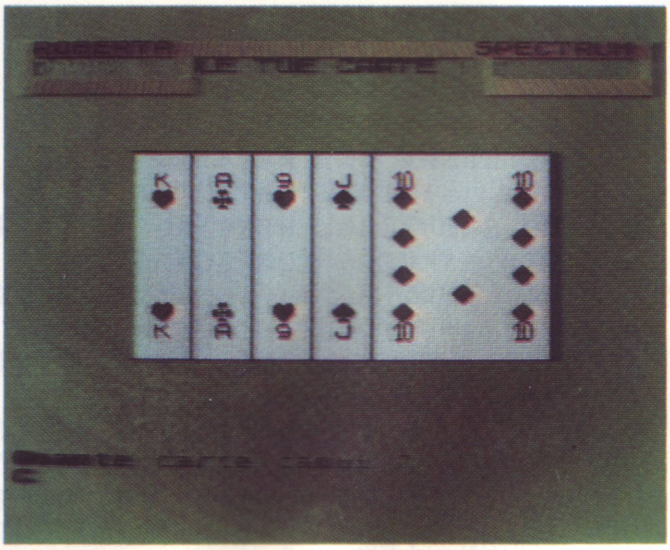
Poker
versione per ZX Spectrum 48K

```
  8 INK 7: CLS : INPUT "Vuoi la
    spiegazione ?(s/n) ";
    LINE g$: IF g$="n" THEN
    GO TO 15
  9 BORDER 1: PAPER 1: INK 6:
    CLS
 10 POKE 23658,0: PRINT AT 0,0;
    "Il gioco e' molto semplice
     si   tratta dello straight
     poker o  poker all'italia
    na."'"Inizialmente distribu
    isco cinquecarte a me,e cin
    que a te."'"Delle mie cinqu
    e mi preoccupo discartare q
    uelle inutili."'"Successiva
    mente sarai tu a doverfare
    lo stesso per te.Tieni ben
    presente che la carta piu'
    a    sinistra e' la numero
    uno ed    andando verso des
    tra il numero  della carta
    aumenta."'"Il capitale iniz
    iale e' di,sia  per me che
    te, 7000 $."'"Per le puntat
    e il minimo e' di  100 $ ed
     il massimo di 3000 $ manon
     cercare di bluffare perche
    '   con me e' tutto inutile.
    "'"In bocca al lupo e vinca
     il     migliore:cioe' io !
    "
 11 LET a$="024779740244202024 7
    7974024422055999774020247 79
    740244220": LET b$="2244624
    4624444<22446244624444<8848
    44444<22446244624444<"
 12 FOR j=1 TO LEN a$: LET n=
    CODE a$(j)-48: LET z=CODE b
    $(j)-48
 13 IF INKEY$<>"" THEN
    GO TO 15
 14 BEEP z/20,n: NEXT j:
    GO TO 11
 15 FOR a=0 TO 39
 20 READ b: POKE USR "a"+a,b:
    NEXT a
 21 DATA 24,60,126,255,255,126,
    24,60
 22 DATA 56,56,16,214,254,214,1
    6,56
 23 DATA 24,60,126,255,255,126,
    60,24
 24 DATA 102,255,255,255,255,12
    6,60,24
 25 DATA 38,105,169,41,41,41,41
    ,118
 26 LET tit=7000: LET tot=7000
 27 POKE 23658,8: POKE 23609,10
    0
 28 INPUT "Come ti chiami ?";z$

 30 BORDER 4: PAPER 4: INK 0:
    CLS
 35 FOR z=4 TO 21: PRINT AT z,0
    ;"
                 ": NEXT z
 37 PRINT AT 1,9;"
    ": PRINT AT 2,9;"
       "
 40 PRINT AT 10,11; FLASH 1;
     PAPER 1; INK 6; BRIGHT 1;"
    ATTENDI !"
 45 REM **DISTRIBUZIONE CARTE
    **
 50 LET j=0: DIM b$ (20,2):
     LET k=0:: DIM t(6): LET d$
    =""
 60 DIM w(8): DIM c$(15,2):
     DIM f(30): LET g=0:
     RANDOMIZE
 70 FOR a=1 TO 19 STEP 2
 80 GO SUB 2000: IF h=1 THEN
     GO TO 80
 90 NEXT a
100 REM **VERIFICA CARTE SPEC
    TRUM**
110 LET pos=0: LET pun=0:
     LET p=0: LET n=0: LET u=0:
     LET z=0
120 FOR a=1 TO 7 STEP 2: FOR d=
    a+2 TO 9 STEP 2
130 IF b$(a)=b$(d) THEN  LET n=
    n+1: LET p=p+1: LET c$(p)=b
    $(a): LET u=INT (a/2+1)
140 NEXT d: NEXT a
150 IF n=0 THEN  GO TO 310
160 IF n=1 OR n=3 THEN  GO TO 5
    60
170 IF n=2 THEN  GO TO 610
190 GO TO 660
300 REM **SCALA COLORE**
310 FOR a=2 TO 8 STEP 2:
     FOR d=a+2 TO 10 STEP 2
320 IF b$(a,1)=b$(d,1) THEN
     LET u=u+1: LET d$=b$(a,1)
330 NEXT d: NEXT a
335 IF u=10 THEN  GO TO 660
340 IF u<6 THEN  GO TO 390
```

LOAD

Poker



```
350 FOR a=1 TO 9 STEP 2: IF b$(
    a+1)=d$ THEN  GO TO 370
360 GO SUB 2000: IF h=1 THEN
    GO TO 360
370 NEXT a: GO TO 660
380 REM **SCALA**
385 FOR a=1 TO 9 STEP 2:
    LET j=INT (a/2+1):
    GO SUB 3000: NEXT a
390 LET p=0: FOR a=1 TO 5:
    FOR d=1 TO 5: IF a=d
    THEN  GO TO 410
400 IF INT f(a)=INT (f(d)-1)
    THEN  LET z=z+1: LET p=p+2
    : LET w(p-1)=INT f(a):
    LET w(p)=INT f(d)
410 NEXT d: NEXT a
420 LET y=1
430 IF z=4 THEN  GO TO 660
440 IF z<3 THEN  GO TO 490
443 REM **NESSUNA COMBINAZION
    E VALIDA**
445 LET q=p: LET p=0
450 FOR a=1 TO 5: LET m=0:
    FOR d=1 TO q: IF m=1
    THEN  GO TO 460
455 IF INT f(a)=w(d) THEN
    LET m=1: LET p=p+1: LET t(
p)=a
460 NEXT d: NEXT a: LET y=9:
    LET j=0
470 FOR a=1 TO 5: LET m=0:
    FOR d=a TO a: IF m=1
    THEN  GO TO 480
475 IF a<>t(d) AND m=0 AND j=
    0 THEN  LET m=1: LET j=1:
    LET y=2*a-1
480 NEXT d: NEXT a: LET a=y
485 GO SUB 2000: IF h=1 THEN
    GO TO 485
487 GO TO 660
490 FOR a=2 TO 5: IF f(a)>f(y)
    THEN  LET y=2*a-1
500 NEXT a
510 FOR a=1 TO 9 STEP 2: IF b$(
    a)=b$(y) THEN  GO TO 530
520 GO SUB 2000: IF h=1 THEN
    GO TO 520
530 NEXT a
540 GO TO 660
550 REM **COPPIA E TRIS**
560 FOR a=1 TO 9 STEP 2: IF c$(
    1)=b$(a) THEN  GO TO 580
570 GO SUB 2000: IF h=1 THEN
    GO TO 570
580 NEXT a
610 FOR a=1 TO 9 STEP 2
630 IF c$(1)=b$(a) OR c$(2)=b$(
    a) THEN  GO TO 650
640 GO SUB 2000: IF h=1 THEN
    GO TO 640
650 NEXT a
660 LET a=9: GO SUB 4000
670 REM **CAMBIO CARTE**
680 INPUT "Quante carte cambi ?
    ", LINE h$: IF h$="" OR h$=
    " " THEN  GO TO 680
685 LET v=VAL h$: IF v<0 THEN
    GO TO 680
687 IF v=0 THEN  GO TO 780
690 IF v>4 THEN  PRINT AT 4,0;
    FLASH 1;"NON PUOI IL MASSI
    MO E'4!": PAUSE 120:
    PRINT AT 4,0;"             ":
    GO TO 680
700 FOR a=1 TO v
705 INPUT "Quale ?", LINE j$:
    IF j$="" OR j$=" " THEN
    GO TO 705
710 LET w(a)=VAL j$
715 IF w(a)<1 OR w(a)>5 THEN
    GO TO 705
717 NEXT a
720 FOR x=1 TO v
730 LET a=(2*w(x)+9)
```

```
740 GO SUB 2000: IF h=1 THEN
  GO TO 740
750 NEXT x
760 LET a=9: GO SUB 4010
770 REM ***** PUNTATA *****

780 INPUT "Quanti dollari punti
 ?"; LINE k$: IF k$=""
 OR k$=" " THEN GO TO 780
783 LET pun=VAL k$: IF pun<100
 THEN PRINT AT 4,0;
 FLASH 1;"NON PUOI IL MINIM
O E' DI 100 $ !": PAUSE 120
: PRINT AT 4,0;"
                     ":
 GO TO 780
784 IF pun>3000 THEN PRINT
AT 4,0; FLASH 1;"NON PUOI I
L MASSIMO E' 3000 $ !":
 PAUSE 120: PRINT AT 4,0;"

      ": GO TO 780
785 IF pun>tot THEN PRINT AT 4
,0; FLASH 1;"NON HAI TANTO
DENARO !": PAUSE 120:
 PRINT AT 4,0;"
                     ":
 GO TO 780
788 IF pun>tit THEN PRINT AT 4
,0; FLASH 1;"NON HO TANTO D
ENARO !": PAUSE 120:
 PRINT AT 4,0;"
                     ":
 GO TO 780
789 REM **VERIFICA COMB.CARTE
**
790 LET y=1: LET b=1: LET m=2:
 LET i=1: LET h=5: LET c=7:
 LET o=8: LET g=9: LET s=10
: GO SUB 795: LET b=11:
 LET y=11: LET c=17: LET g=
19: LET h=15: LET o=18:
 LET s=20: LET m=12: LET i=
2: GO SUB 795: GO TO 1052
795 LET q=0: LET l=0: LET n=0:
 LET p=0: LET z=0: LET u=0
800 FOR a=b TO c STEP 2: FOR d=
a+2 TO g STEP 2
810 IF b$(a)=b$(d) THEN LET n=
n+1: LET p=p+1: LET q=p:
 LET c$(q)=b$(a)
820 FOR r=1 TO q-1: IF c$(q)=c$
 (r) THEN LET p=p-1: IF p<0
 THEN LET p=0
830 NEXT r
832 NEXT d: NEXT a
833 IF c$(1)=c$(2) OR n<>4
  THEN GO TO 851
835 FOR d=2 TO 3: IF c$(1)=c$(d
) THEN GO TO 845
840 LET c$(4)=c$(1): LET c$(1)=
c$(d): LET c$(d)=c$(4)
845 NEXT d
850 IF n=4 THEN LET n=5
851 IF n>=1 THEN GO TO 975
853 IF n>0 THEN GO TO 975
855 FOR a=1 TO 19 STEP 2: IF a>
10 THEN LET k=5
856 LET j=INT (a/2+1+k)
857 GO SUB 3000: NEXT a
858 REM ***VERIFICA SCALA***

860 FOR a=b TO h: FOR d=b TO h:
  IF a=d THEN GO TO 880
870 IF INT f(a)=INT (f(d)-1)
  THEN LET z=z+1
880 NEXT d: NEXT a
890 FOR e=m TO h: IF f(e)>f(y)
  THEN LET y=e
900 NEXT e
905 REM **VERIFICA SCALA COLO
RE**
910 FOR a=m TO o STEP 2: FOR d=
a+2 TO s STEP 2
920 IF b$(a,1)=b$(d,1) THEN
  LET u=u+1: LET d$=b$(a,1)
930 NEXT d: NEXT a
940 IF u=10 AND z=4 THEN LET n
=8+(CODE d$/1000): GO TO 10
50
945 IF u=10 THEN LET n=7:
  GO TO 1050
950 IF z=4 AND INT f(y)=11
  THEN LET n=4.2: GO TO 105
0
960 IF z=4 AND INT f(y)=12
  THEN LET n=4.4: GO TO 105
0
965 IF z=4 AND INT f(y)=13
  THEN LET n=4.5: GO TO 105
0
970 IF z=4 AND INT f(y)=14
  THEN LET n=4.6: GO TO 105
0
973 IF n>=1 THEN LET l=f(y):
  GO TO 1030
975 LET a=1: GO SUB 980: LET x=
1: IF n=2 THEN LET a=2:
  GO SUB 980: LET l=l+x
978 GO TO 1040
```

LOAD

Poker

```
 980 IF c$(a,1)="E" THEN  LET l=
     10: GO TO 1030
 985 IF c$(a,1)="J" THEN  LET l=
     11: GO TO 1030
 990 IF c$(a,1)="Q" THEN  LET l=
     12: GO TO 1030
1000 IF c$(a,1)="K" THEN  LET l=
     13: GO TO 1030
1010 IF c$(a,1)="A" THEN  LET l=
     14: GO TO 1030
1015 IF c$(a)="  " THEN  GO TO 1
     030
1020 LET l=VAL c$(a)
1030 LET l=l/100: RETURN
1040 LET n=n+1
1050 LET t(i)=n: RETURN
1052 IF pun>1500 AND t(2)<3
      THEN  LET g=2
1055 LET m=0: LET v=0: IF INT t(
   1)=0 AND tit>=pun+1000
   AND tot>=pun+1000 THEN
   PRINT AT 4,0;"RILANCIO DI
  1000 !               ":
   LET g=1: GO TO 1072
1058 IF g=2 THEN  GO TO 1090
1060 IF pun>1500 AND t(1)<3
     AND t(2)>3 THEN  PRINT
     AT 4,0;"PASSO !
                "": PAUSE 150:
     LET m=1: GO TO 1130
1070 IF t(1)>4 AND tit>=pun+1000
     AND tot>=pun+1000 THEN
     PRINT AT 4,0;"RILANCIO DI
     1000 !            ":
     GO TO 1072
1071 GO TO 1090
1072 LET v=1000: INPUT "COSA FAI
     ?", LINE d$: IF d$="PASSO"
     THEN  LET m=1: GO TO 1120
1075 IF d$="VEDO" AND tot<pun+v
     THEN  PRINT AT 5,0;
     FLASH 1;"NON HAI TANTO DEN
     ARO !": PAUSE 150: PRINT
     AT 5,0;"
                 ": GO TO 1120
1080 IF d$="VEDO" THEN  GO TO 11
     00
1085 GO TO 1072
1090 PRINT AT 4,0;"VEDO !
                           ":
     PAUSE 150
1095 REM **ASSEGNAZIONE VINCIT
     ORE**
1100 IF t(1)=t(2) THEN  PRINT
     AT 4,0;"LA MANO E' PARI !
              "'"SEI UN VALID
     O AVVERSARIO.         ":
      GO TO 1140
1110 IF t(1)<t(2) THEN  GO TO 11
     30
1120 PRINT AT 4,0;"HO VINTO IO.S
     ONO TROPPO FORTE.   ": LET t
     ot=tot-pun-v: LET tit=tit+p
     un+v: IF v=0 AND m=1 THEN
      LET tot=INT (tot+pun-pun*1
     0/100): LET tit=INT (tit-pu
     n+pun*10/100)
1125 GO TO 1140
1130 PRINT AT 4,0;"HAI VINTO TU
     !                         ":
      PRINT "COMPLIMENTI.
                    ": LET tot=tot
     +pun+v: LET tit=tit-pun-v:
      IF v=0 AND m=1 THEN  LET t
     ot=INT (tot-pun+pun*10/100)
     : LET tit=INT (tit+pun-pun*
     10/100)
1140 IF g=2 THEN  PRINT AT 18,0;
     "HAI BLUFFATO !": GO TO 114
     4
1143 IF g=1 THEN  PRINT AT 18,0;
     "HO BLUFFATO !"
1144 LET a=-1: GO SUB 4020
1145 IF tit<100 THEN  CLS :
      PRINT AT 0,0; FLASH 1;"NO
     N HO PIU' DENARO PER GIOCAR
     E !": LET w(1)=6: GO TO 116
     0
1146 IF tot<100 THEN  CLS :
      PRINT AT 10,0; FLASH 1;"NO
     N HAI PIU' DENARO PER GIOCA
     RE!": LET w(1)=6: GO TO 116
     0
1160 INPUT "Vuoi continuare ?(S/
     N) "; LINE p$: IF p$="N"
     THEN  STOP
1165 IF w(1)=6 THEN  GO TO 26
1170 GO TO 35
2000 REM ***GENERAZIONE CARTE*
     **
2001 LET h=0: LET t=2: LET g=g+1
     : LET b=INT (RND*6)+9:
     LET c=RND*.004+.144: LET b
```

```
    =b+(INT (c*1000))/1000:
     IF INT b=10 THEN  LET b$(a
    )="E": GO TO 2040
2005 IF INT b=11 THEN  LET b$(a)
    ="J": GO TO 2040
2010 IF INT b=12 THEN  LET b$(a)
    ="Q": GO TO 2040
2020 IF INT b=13 THEN  LET b$(a)
    ="K": GO TO 2040
2030 IF INT b=14 THEN  LET b$(a)
    ="A": GO TO 2040
2035 LET b$(a)=STR$ b
2040 IF c<.146 THEN  LET t=0
2050 LET f(g)=b: LET s=g
2060 FOR e=1 TO s-1: IF f(e)=b
     THEN  LET g=g-1: LET h=1
2070 NEXT e
2080 LET b$(a+1,1)=CHR$ INT (c*1
    000): LET b$(a+1,2)=STR$ t
2090 RETURN
3000 IF b$(a)="E " THEN  LET f(j
    )=10: RETURN
3005 IF b$(a)="J " THEN  LET f(j
    )=11: RETURN
3010 IF b$(a)="Q " THEN  LET f(j
    )=12: RETURN
3020 IF b$(a)="K " THEN  LET f(j
    )=13: RETURN
3030 IF b$(a)="A " THEN  LET f(j
    )=14: RETURN
3040 LET f(j)=VAL b$(a):
     RETURN

3990 REM *****PARTE GRAFICA***
    **
4000 FOR z=4 TO 21: PRINT AT z,0
    ;"            ": NEXT z: PRINT
    AT 1,9; INK 0;"LE TUE CARTE
    !": GO TO 4040
4010 PRINT AT 1,9; INK 0;"QUESTE
     SONO LE": PRINT AT 2,10;"N
    UOVE CARTE!": FOR z=1 TO v:
     IF w(z)=5 THEN  GO TO 4030
4011 NEXT z: GO TO 4040
4020 PRINT AT 1,9; INK 0;" QUEST
    E ERANO ": PRINT AT 2,10;
     INK 0;"LE MIE CARTE!":
     GO TO 4030
4030 PRINT AT 9,19; PAPER 7;
     BRIGHT ;"      ":
     PRINT AT 10,19; PAPER 7;
     BRIGHT 1;"      ":
     PRINT AT 11,19; PAPER 7;
     BRIGHT 1;"      ":
     PRINT AT 12,19; PAPER 7;
     BRIGHT 1;"      ":
     PRINT AT 13,19; PAPER 7;
     BRIGHT 1;"      ":
     GO TO 4044
4040 PRINT AT 0,0; INK 6;"
     {32SG8}"
4041 PRINT AT 1,0; INK 6;"{SG8}"
    : PRINT AT 1,8; INK 6;"
     {SG8}": PRINT AT 1,23;
     INK 6;"{SG8}": PRINT AT 1,
     31; INK 6;"{SG8}"
4042 PRINT AT 2,0; INK 6;"
     {9SG8}": PRINT AT 2,23;
     INK 6;"{9SG8}"
4043 PRINT AT 0,1; PAPER 2;
     INK 6; OVER 1;z$: PRINT
    AT 0,23; PAPER 2; INK 6;
     OVER 1;"SPECTRUM"
4044 PRINT AT 1,2;"      ":
     PRINT AT 1,25;"      "
4045 PRINT AT 1,1; INK 7;tot;" $
    ": PRINT AT 1,24; INK 7;tit
    ;" $": IF z=22 THEN  LET z=
    0: GO TO 4049
4046 GO TO 4085
4049 FOR y=6 TO 16
4050 PRINT AT y,6; PAPER 7;
     BRIGHT 1;"
          "
4060 NEXT y
4070 PLOT 215,127: DRAW -168,0:
     DRAW 0,-88: DRAW 168,0
4080 FOR x=47 TO 143 STEP 24:
     PLOT x,127: DRAW 0,-88:
     NEXT x: PLOT 215,127:
     DRAW 0,-88
4085 LET q=a
4090 FOR c=7 TO 15 STEP 8: LET a
    =q
4100 FOR b=7 TO 19 STEP 3: LET a
    =a+2
4110 PRINT AT c,b; PAPER 7;
     BRIGHT ; INK VAL b$(a+1,2
    );b$(a,1)
4120 PRINT AT 8,b; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
    );b$(a+1,1)
4130 PRINT AT 14,b; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
    );b$(a+1,1)
4140 NEXT b
4160 PRINT AT 8,25; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
    );b$(a+1,1)
```

```
4170 PRINT AT c,25; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );b$(a,1)
4180 PRINT AT 14,25; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );b$(a+1,1)
4190 NEXT c
5020 IF b$(a,1)="9" THEN
     GO TO 5300
5030 IF b$(a,1)="E" THEN
     GO TO 5400
5040 IF b$(a,1)="J" THEN
     GO TO 5500
5050 IF b$(a,1)="Q" THEN
     GO TO 5600
5060 IF b$(a,1)="K" THEN
     GO TO 5700
5070 IF b$(a,1)="A" THEN
     GO TO 5800
5300 PRINT AT 10,19; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );b$(a+1,1)
5310 PRINT AT 10,25; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );b$(a+1,1)
5320 PRINT AT 11,22; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );b$(a+1,1)
5330 PRINT AT 12,19; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );b$(a+1,1)
5340 PRINT AT 12,25; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );b$(a+1,1): RETURN
5400 PRINT AT 10,19; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );b$(a+1,1)
5410 PRINT AT 10,25; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );b$(a+1,1)
5420 PRINT AT 9,22; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );b$(a+1,1)
5430 PRINT AT 12,9; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );b$(a+1,1)
5440 PRINT AT 12,25 PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );b$(a+1,1)
5450 PRINT AT 13,22; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );b$(a+1,1): RETURN
5500 PRINT AT 9,21; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );"{3SG3}"
5510 PRINT AT 10,22; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );"{G5}"
5520 PRINT AT 11,22; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );"{G5}"
5530 PRINT AT 11,22; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );"{G5}"
5540 PRINT AT 12,21; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );"{SG1}{SG2}": RETURN
5600 PRINT AT 9,21; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );"{3SG3}"
5610 PRINT AT 10,21; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );"{SG5} {G5}"
5620 PRINT AT 11,21; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );"{SG5} {G5}"
5630 PRINT AT 12,21; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );"{SG1}{SG3}{SG8}"
5640 PRINT AT 13,24; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );"{G2}": RETURN
5700 PRINT AT 9,21; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );"{G5}"
5710 PRINT AT 0,21; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );"{G5} {SG6}"
5720 PRINT AT 11,21; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,2
     );"{G5}{SG6}"
5730 PRINT AT 12,21; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,
     );"{G5}{G6}"
5740 PRINT AT 13,21; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,
     );"{G5} {G6}": RETURN
5800 PRINT AT 9,21; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1,
     );"{3SG3}"
5810 PRINT AT 10,21; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1
     );"{SG5} {G5}"
5820 PRINT AT  ,21; PAPER 7;
     BRIGHT 1; INK VAL b$(a+1
     );"{SG1}{SG3}{SG2}"
5830 PRINT AT 12,21; PAPER 7;
     BRIGHT 1; INK VAL b$(a+
     );"{SG5}{G5}"
5840 PRINT AT 13,21; PAPER 7;
     BRIGHT 1; INK VAL b$(a+
     );"{SG5} {G5}": RETURN
```

UDG: ovvero l'Utilizzo Dettagliato (dei caratteri) Grafici

di **Carlo Panzalis**

parte prima

Una piccola ma esauriente guida all'uso dei caratteri grafici dello Spectrum



Come è noto lo Spectrum, sia nella versione a 16 che a 48 Kbyte di RAM, consente all'utente di definire 21 caratteri grafici a proprio piacimento, ciascuno dei quali associato ad una lettera da A ad U.
Tuttavia molto spesso l'utente stenta a prendere rapidamente confidenza, con questa peculiarità dello Spectrum, in modo da sfruttarla adeguatamente per arricchire i propri programmi.
È proprio pensando a questo "lettore in difficoltà" che iniziamo questa carrellata sulle caratteristiche del set di caratteri, sia residenti sia definibili dall'utente, nonché sulla gestione dell'immagine video, nella speranza di riuscire a rispondere ai più tipici interrogativi relativi a questo tema, fra i quali: come definire i caratteri grafici; come utilizzarli in un programma; come superare il tetto di 21 caratteri e, ad esempio, definire un nuovo set completo di caratteri alfanumerici da utilizzare al posto di quello residente, e così via.
Innanzitutto è opportuno ricordare che lo schermo preenta 22 linee (orizzontali) e 32 colonne (verticali), e risulta quindi suddiviso in 22*32 = 704 quadratini. Onguno di questo, che indicheremo genericamente con il nome carattere, è composto da 64 pixel (punti), ognuno dei quali può essere acceso o spento (cioè avere stato 1 o 0), assumendo così rispettivamente il colore dell'inchiostro (INK) o il colore della carta (PAPER). Sono dunque le numerosissime combinazioni di carta ed inchiostro a determinare l'aspetto dei diversi caratteri. Quando sullo schermo non ci sono né scritte né altri simboli, ogni carattere è costituito da 64 pixel a stato 0, vale a dire tutti del colore della carta; se decidete di stampare la lettera A nel quadratino in alto a sinistra (coordinate 0,0) in quest'ultimo alcuni pixel si accenderanno assumendo il colore dell'inchiostro e consentendo la rappresentazione del simbolo prescelto. Tutti i simboli imprimibili sul video, siano essi caratteri residenti, simboli grafici sulla prima fila di tasti o quelli definibili dall'utente, sono dunque costituiti da 8 righe orizzzontali di 8 pixel ciascuna, e possono essere stampati solo in corrispondenza di uno dei 704 quadratini di cui sopra.
A scopo esemplificativo costruiamo una griglia di 8*8 quadratini di lato, ognuno dei quali rappresenta naturalmente un pixel, e reppresentiamo l'aspetto del carattere Spectrum A maiuscolo. Come è agevole notare, la prima e l'ultima riga di 8 pixel sono costituite da pixel spenti (per realizzare la separazione dei caratteri verso l'alto e il basso) e quindi tutti a stato 0. Procedendo dall'alto verso il basso vediamo invece che nella seconda riga quattro pixel sono accesi ed assumeranno quindi il colore dell'inchiostro. Il numero binario 00111100 (60 convertito in sistema decimale) potrà dunque rappresentare esattamente questa situazione. In modo del tutto analogo possono essere rappresentate attraverso codici numerici le altre righe di pixel. Gli 8 numeri che codificano il simbolo A sono contenuti negli 8 byte (gruppi di 8 bit, in questo caso rappresentanti lo stato dei pixel di un carattere) dalla locazione 15880 alla 15887: ogni volta che volete imprimere la lettera A lo Spectrum legge il contenuto di queste locazioni e lo trasferisce nelle locazioni che gestiscono l'immagine video (Display File), cui è devoluto il compito di decodificare i numeri in combina-

zioni di pixel accesi/spenti. Tutto il set dei caratteri residenti (dallo spazio al segno di copyright), è contenuto secondo quanto ora descritto a partire dalla locazione 15616 alla 16383. Si tratta di 96 simboli, ognuna dei quali occupa in memoria (ROM) 8 byte.
Compreso questo meccanismo, il lettore non dovrebbe avere difficoltà a definire un qualsiasi simbolo grafico, individuando i codici numerici (in notazione binaria o decimale indifferentemente) atti a rappresentarlo: si tratterà sempre di disegnare una qualsiasi figura in una matrice di 8*8 punti, trovare gli 8 numeri ad essa relativi e collocarli nelle idonee locazioni di memoria. A tale scopo nello Spectrum sono riservati 168 byte (21*8) nella RAM, più precisamente gli utlimi 168 byte fisicamente presenti. Stante la differente capacità della memoria ad accesso casuale nelle due versioni di questa macchina, troveremo che nel 16K sono a questo scopo riservati i bytes della locazione 32600 alla 32767, mentre nel 48K i butes in questione occupano gli indirizzi da 65368 a 65535. All'accensione questi bytes contengono i valori numeri delle prime 21 lettere dell'alfabeto (inglese): se entrare in modo grafico (CAPS SHIFT e 9) e premete i tasti A — U vedrete apparire sullo schermo le corrispondenti lettere maiuscole. Tuttavia il contenuto di queste locazioni può essre modificato dall'utente permettendo la rappresentazione di un qualsiasi simbolo grafico. Supponendo di avere già trovato gli 8 codici numerici relativi ad un simbolo grafico idoneo ai nostri scopi, potremmo attraverso l'istruzione POKE collocare tali valori nelle prime 8 locazione di questa area, associando il simbolo al tasto 'A' nel seguente modo: (Es. riferito al 16K)

```
POKE 32600, numero
POKE 32601, numero
POKE 32602, numero
ecc. fino a
POKE 32607, numero
```

Infatti POKE permette di 'forzare' un byte e di fargli assumere un qualsiasi valore compreso tra 0 e 255, indipendentemente dal valore che esso conteneva. Entrando in modo grafico (premendo CAPS SHIFT e 9 simultaneamente) e, quindi premendo il tasto 'A', vedremo realizzato sullo schermo il simbolo da noi costruito.
Operare in questo modo presenta tuttavia alcuni inconvenienti. Innanzitutto si tratta di un procedimento piuttosot lungo: infatti dovremo calcolare le locazioni in cui collocare i valori in relazione ai differenti tasti cui lo vogliamo associare ed inoltre dovremo scrivere le 8 istruzioni necessarie (gli 8 POKE), e questo per ogni carattere da noi desiderato. Per migliorare tale procedura sarà sufficiente predisporre una linea di DATA, composta da 8 dati, per ogni simbolo grafico e realizzare quindi un ciclo FOR... NEXT che legga tali dati e li collochi nelle locazioni desiderate:

```
FOR n = 0 TO 7
READ dato
POKE 32600 + n, dato
NEXT n
DATA 0,12,60,255,1,40,190,81
```

(i dati sono del tutto casuali)

In questo loop grazie alla variabile n ad ogni passaggio la locazione considerata si incrementa di 1, caricando il numero ad essa relativo.
Anche in questo caso si manifesta tuttavia la necessità di calcolare le locazioni interessate al procedimento in relazione al particolare tasto cui esse sono associate: così abbiamo infatti associato un simbolo grafico al tasto 'A', mentre per associarlo al tasto 'R' dovremmo calcolare gli indirizzi ad esso relativi. Inoltre così facendo i listati per il 16 e il 48K perderebbero la loro compatibilità poiché nelle due macchine diverse sono le aree RAM riservate agli UDG (a proposito: tale sigla significa caratteri grafici definibili dall'utente — User-Defined Graphics). Per scavalcare tali problemi è necessario ricorrere alla funzione USR. Questa è una funzione che, applicata ad una stringa composta da una lettera da A ad U, ritorna l'indirizzo del primo byte degli otto cui la lettera è associata. Ad esempio se inserite come comando immediato:

```
PRINT USR"a"
```

otterrete 32600 nel 16K, oppure 65368 nel 48K.
Quindi l'istruzione

```
POKE USR "a", numero
```

è del tutto equivalente a

POKE 32600, numero nel 16K, ed invece a

POKE 65368, numero nel 48K
facendo così salva la compatibilità dell'operazione nelle due versioni dello Spectrum. Sarà dunque sufficiente sostituire nel loop di cui sopra la linea del comando POKE con una del tipo:

```
POKE USR "a" + n, dato
```

Inoltre per associare il simbolo in questione a una qualsiasi lettera (A-U) basterà sostituire la stringa "a" con una contenente la lettera desiderata, eliminando così ogni noioso calcolo. Il procedimento in questione è particolarmente utile in caso dobbiate definire, come di norma avviene, diversi caratteri grafici. In questo caso il loop che carica in memoria i codici numerici dovrà essere nidificato in un altro che tenga conto del numero di caratteri da definire. Un esempio classico è:

```
FOR f = 1 TO 3
READ p$
FOR n = 0 TO 7
READ dato
POKE USR p$ + n, dato
NEXT n
NEXT f
DATA "a",11,25,255,0,0,0,133,14
DATA "g",0,0,100,98,76,45,254,98
DATa "t",13,14,56,199,193,1,0,1
```

Il loop esterno legge il primo dato stringa ("a"), mentre il loop interno legge uno ad uno i primi 8 dati numerici e li carica nelle locazioni associate alla lettera A; poi il loop esterno legge la seconda stringa ("g") ed il loop interno carica i byte associati alla lettera G; e così via.
Il loop esterno può essere costruito in diverse forme alternative ma equivalenti in relazione alle diverse esigenze dell'utente. Ad esempio, se si devono definire 11 caratteri grafici e si intede associarli alle prime 11 lettere (A-K), si può far riferimento al codice decimale della lettera nel seguente modo:

```
FOR f = 144 TO 151
....
POKE USR CHR$ f + n, dato
....
NEXT f
```

evitando così di dover riportare all'inizio di ogni linea di DATA la lettera cui i dati stessi si riferiscono.
Tale procedura non risulta invece efficiente nel caso si intendono associare i caratteri ad un insieme di lettere non legate da una relazione così lineare (ad esempio un gruppo di lettere quali a, b, j, r, t, u....).
Una volta definiti dei caratteri grafici, li potremo sfruttare nell'ambito di una programma utilizzando l'istruzione PRINT seguita da una stringa contenente le lettere associate ai simboli stessi. Se, per esempio, abbiamo definito un omino in corrispondenza del tasto 'S' e lo vogliamo imprimere al centro dello schermo, basterà inserire nel listato una linea così:

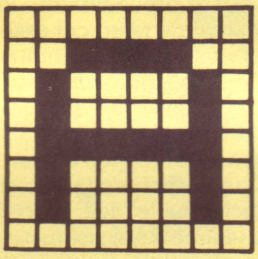
```
PRINT AT 11,15; "S"
```

La stringa deve però essere necessariamente battuta in modo grafico, vale a dire che dopo aver aperto gli apici dovremo premere il tasto CAPS SHIFT e il tasto 9: subito il cursore dalla stato 'L' passerà allo stato grafico 'G'. Premendo il tasto 'S' identificheremo il carattere da rappresentare, quindi ripremendo il tasto 9 potremo chiudere gli apici. Quando nel corso dell'esecuzione del programma si giungerà alla linea della PRINT, vedremo apparire sullo schermo il nostro omino secondo le specifiche da noi caricate in memoria. Inoltre prmendo ENTER vedremo che la lettera è stata sostituita anche nel listato dal simbolo grafico ad essa assegnando. Questa particolarità ci consente di verificare in una data posizione dello schermo il carattere giusto.

Per immagazzinare dei caratteri da noi definiti all'interno di un programma dovremo dunque realizzare i due loop atti a caricare i dati nelle locazioni di memoria ad essi destinate. È comunque essenziale che, prima di essere stampati sullo schermo, i caratteri vengano caricati in memoria: la subroutine di caricamento, costituita dai due loop, deve quindi essere attivata all'inizio del programma, eventualmente richiamandola con una istruzione GO SUB.
Segnaliamo comunque che, a causa dell'elevato numero di byte richiesti da ogni linea DATA, è in taluni casi opportuno dividere il nostro progamma in due listati: il primo sarà destinato solo al caricamento dei caratteri grafici; il secondo sarà invece il programma vero e proprio. Questa possibilità è realizzabile anche grazie al fatto che l'area destinata agli UDG è posta al di sopra di RAMTOP; con questo termine si indica la soglia massima di RAM disponibile per la programmazione BASIC. Tutto ciò che si trova al di sopra di tale soglia è protetto sia nel caso si carichi un nuovo programma (nel nostro caso il listato 2, cioé il nostro programma 'vero'), sia nel caso si dia un comando NEW. Provate a definire un carattere grafico, quindi battete NEW: entrando in modo grafico vedrete che il carattere è ancora in memoria, mentre è sparito il loop con cui lo avevate definito.
Quanto detto dovrebbe avervi suggerito una terza possibilità di caricamento degli UDG: una volta definiti nel solito modo è possibile salvarli su nastro con il comando immediato

```
SAVE "nome" CODE USR "a", 168
```

dove "nome" va sostituito con qualsiasi nome (meno di 10 lettere) voi vogliate.
Quando vorrete caricare il vostro programma dovrete caricare con LOAD "" CODE anche questi byte, ottenendo il risultato di avere caratteri grafici senza avere il listato appesantito alla subroutine di caricamento.
Sperando di aver reso più agevole l'accesso al tema in questione all'utente insoddisfatto delle spiegazioni del manuale, necessariamente rapide e motivo della vastità degli argomenti trattati, rinviamo ad una prossima puntata l'esame di particolari problemi connessi alla creazione di caratteri grafici attraverso tecniche alternative, capaci di superare alcuni vincoli propri del procedimento qui mostrato.



BIN	00000000	=	0	(decimale)
BIN	00111100	=	60	(decimale)
BIN	01000010	=	66	(decimale)
BIN	01000010	=	66	(decimale)
BIN	00111110	=	126	(decimale)
BIN	01000010	=	66	(decimale)
BIN	01000010	=	66	(decimale)
BIN	00000000	=	0	(decimale)

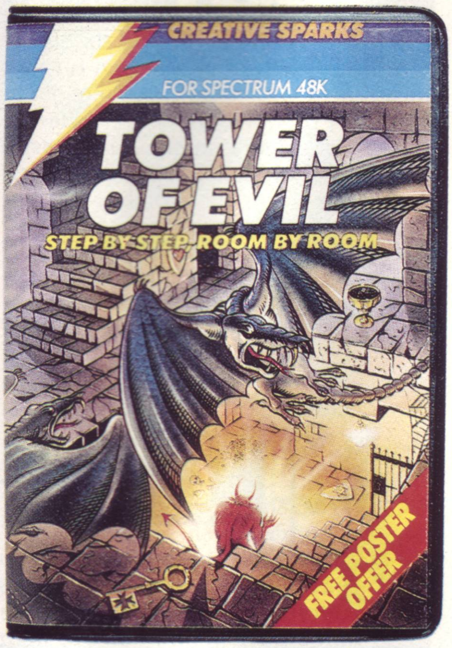
Gli otto valori della tabella sottostante sono i codici numerici in notazione binaria e decimale che consentono la rappresentazione sul video del carattere A maiuscolo. Al momento della accensione dello Spectrum detti valori sono contenuti in otto locazioni consecutive a partire dalla 15880, nonché in altre otto a partire dalla 32600 (65368 nel 48K).
I primi otto indirizzi fanno parte dell'area ROM in cui è immagazzinato il set di caratteri residenti, e quindi il loro contenuto non può essere modificato.
Gli altri otto bytes fanno invece parte della sezione RAM destinata a contenere gli UDG: queste locazioni sono accessibili all'utente e quindi il loro contenuto può essere modificato così da realizzare caratteri grafici personalizzati.

Recensioni Software

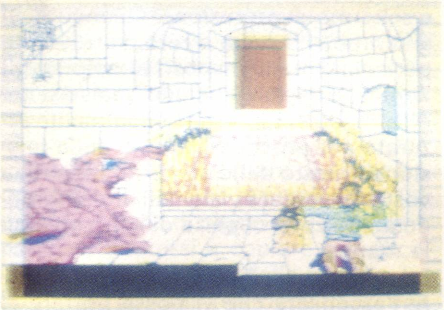
a cura di
Lucio Bragagnolo

Tower of Evil

Avete il coraggio di avventurarvi nella torre del male?

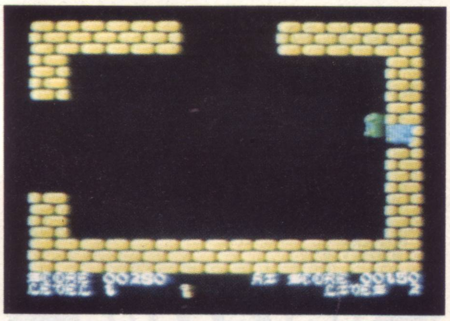


Questo gioco si inserisce nel filone, tanto per fare un nome, di Atic Atac: una rischiosissima ricerca all'interno di un castello - nel nostro caso una torre - per trovare qualche cosa (i tre pezzi della chiave che consente di aprire il portone e di uscire dal castello, nell'esempio che abbiamo appena fatto - lo sapevate?): in questo caso, ci caliamo nei panni di Andros, eroe cacciato dalla corte di re Salimos e condannato a non fare mai più ritorno fino a che non avrà recuperato il tesoro del re, trafugato dal malvagio Negromante. L'unico suo aiuto è un incantesimo generosamente elargitogli dal mago di



corte, che gli consentirà di scagliare palle di fuoco dall le punte delle dita. All'interno della Torre del Male sono sparsi i nove tesori da recuperare, più le chiavi che permettono il passaggio al livello successivo. Ovviamente non è tutto qui: le varie stanze della torre sono popolate dai servi del Negromante, che faranno del loro meglio per complicare le cose.

I livelli da superare sono cinque, e non sono affatto facili, anche perché i tesori conquistati saranno al sicuro solamente una volta messi nel baule che appare nella stanza di partenza, al primo piano, obbligando Andros a continui andirivieni tra un piano e l'altro. Ogni volta infatti che il nostro eroe perderà una vita, tutti i tesori in quel momento in suo possesso verranno persi e sarenno nuovamente da recuperare. Le vite a disposizione sono cinque ma... non bastano proprio! Oltre agli infortuni derivati da incapacità o distrazione capita spessissimo di venire colpiti praticamente a tradimento nel passaggio tra una stanza e l'altra; oltretutto per passare da un livello all'altro occorre superare un insidiosissimo corridoio di fuoco interrotto più volte da pozze di lava bollente, che occorre superare con balzi di discreta precisione, pena la morte per liquefazione rapida (uno dei piccoli ma grade-



voli effetti speciali che il programmatore ha inserito, pur nel contesto di una grafica piuttosto spartana).

Il gioco è praticamente tutto qui, basato più sulla velocità e sulla precisione di chi gioca piuttosto che sulla spettacolarità della grafica (comunque adeguata) o sulla complessità del meccanismo di gioco.

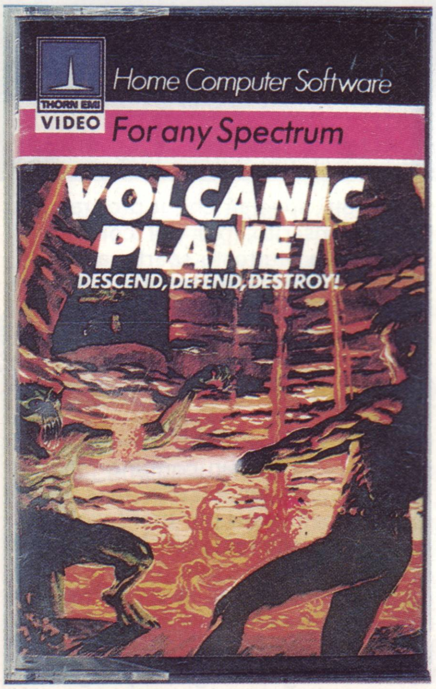
Non crediate però che Tower of Evil sia null'altro che un comune "spara e fuggi": senza un pizzico di capacità strategiche i vostri pur meritevoli sforzi non verranno mai premiati in misura adeguata. Occorre infatti in primo luogo saper ottimizzare la spola tra i vari livelli e il forziere contenuto nel primo, per evitare viaggi inutili e rischiosi; in secondo luogo, bisogna saper girare il meno possibile per ogni piano restando contemporaneamente in ogni stanza per il minor tempo possibile, perché il numero di "cattivi" presenti nel livello in cui siete dipende dal vostro tempo di permanenza e dal numero di ulteriori passaggi in una stanza già precedentemente visitata. Un suggerimento: non abbiate paura di sprecare colpi, visto che il fuoco continuo protegge entro certi limiti la vostra avanzata e, molte volte, vi salva da situazioni alle quali i vostri riflessi non avrebbero potuto in nessun caso rispondere in tempo utile. Cercate anche di essere veloci

R.U.N

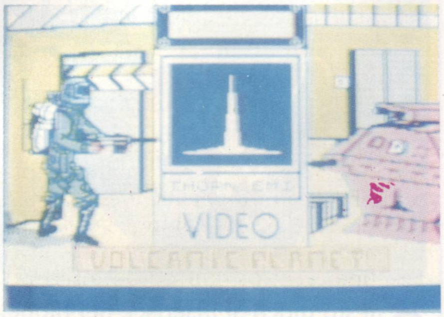
- più tempo vi fermate e più rischi correte; se proprio dovete fermarvi, mai farlo vicino a una porta!
Un accenno, infine, alla confezione, discretamente elegante e pratica, e alla documentazione, dettagliatissima ed esauriente.
Non vi pentirete certo dell'acquisto di Tower of Evil: è un gioco che saprà impegnarvi al massimo delle vostre possibilità, senza stancare od annoiarvi.

Volcanic Planet

riuscirete ad avere ragione dei perfidi Zerans?



State per calarvi nei panni di un sabotatore, incaricato di addentrarvi nelle viscere del pianeta Zeras, abitato dai malvagi Zerans, per minarne il centro (che — non sembrano molto furbi, questi alieni — poggia direttamente sopra un immane vulcano) e provocare in questo modo l'inondazione di un oceano di lava che spazzi via per sempre il pianeta stesso insieme ai suoi abitanti.
Per compiere la missione, partite dalla grande metropoli sotterranea che sorge su Zeras e, forniti di tutto l'occorrente — ossigeno, sistema di mantenimento, energia e armamento, quest'ultimo consistente in un micidiale blaster — vi introducete nella città. Gli Zerans non presteranno molta attenzione all'intruso, almeno inizialmente.
In alto a destra potete vedere una mappa, forse un po' piccola rispetto al necessario, del livello in cui vi trovate. La vostra posizione è in essa sengata dal lampeggiare di un cursore color magenta; sotto la mappa si trovano invece gli indicatori dell'efficienza delle vostre apparecchiature, display (cioè la capacità di evidenziare la vostra posizione sulla mappa generale), bio-system, energia e schermo di protezione. Quando quest'ultimo si troverà a zero, vi troverete pericolosamente esposti ai colpi degli Zerans, e basterà che uno qualsiasi degli altri indicatori venga ugualmente azzerato per provocare la fine della missione. Il resto dello schermo mostra la vostra avventura vista dall'alto, ingrandendo la zona coperta dal cursore sulla mappa.
La prima cosa che dovete fare è cercare l'ascensore che vi possa portare la piano inferiore. Gli ascensori da cercare saranno uno per piano, in quanto essi fanno la spola continuamente tra un piano e un altro: saprete di essere nelle vicinanze di uno di essi osservando il cursore sulla mappa diventare di color ciano (guardate sulla tastiera dello Spectrum se non conoscere questo termine, più precisamente il tasto "5"). Non c'é modo di sapere se l'ascensore che avete trovato sale o scende: l'unico sistema è provare, verificando l'esito del vostro tentativo su un indicatore di profondità visibile in alto a sinistra. Tenendo anche presente che entrare nel primo ascensore, quello da cui partite, consente un reintegramento immediato delle voste risorse. I livelli da cui la città è composta variano da 3 a 15, a seconda del livello di difficoltà prescelto, e dovrete quindi trovare altrettanti ascensori. Nel frattempo avrete già avuto il vostro da fare: difatti la pianta di ogni livello è estremamente arzigogolata, e trovare un ascensore non è facile; in più, gli Zerans diventeranno sempre più intraprendenti e bellicosi, coinvolgendovi in terrificanti conflitti a fuoco e costringendovi a volte a cimentarvi in fughe degne dell'Harrison Ford di "Blade Runner" o dei "Predatori dell'Arca perduta". Tra l'altro, appena uno Zeran viene colpito gli altri aumentano vertiginosamente velocità e casualità del movimento per qualche istante, creandovi ulteriori difficoltà. Cosa manca?... Ah, si: nella città si trovano sparsi alcuni kit di riparazione, che in certi casi si dimostrano davvero utili.
Arrivati finalmente sul fondo e localizzata la bocca del vulcano, non vi resta che depositare la bomba che avete portato con voi, e darvi alla fuga nel tentativo di ritrovare l'uscita nel più breve tempo possibile. La bomba esploderà dopo 15 secondi, e la lava fusa comincerà ad invadere il sotterraneo, arrostendo i malcapitati Zerans e, se non sarete stati abbastanza rapidi, anche voi.
VOLCANIC PLANET è un gioco di azione, che richiede però una buona dose di strategia, senza la quale siete destinati a vagare senza meta per i sotterranei fino alla morte per consumazione.
È decisamente impegnativo da giocare con la tastiera, mentre con un joystick ci si trova sicuramente meglio: come tutti i prodotti Thorn Emi che abbiamo visto finora, anche questo assicura la compatibilità con i joystick Kempston e Sinclair. Considerato anche che il programma gira anche su Spectrum inespanso da 16K, si può parlare di prodotto pervettamente proporzionato alle aspettative, in grado di entusiasmare gli amanti del genere.



Tower of Evil
(cod. DTEMSO2)
L. 16.000

Volcanic Planet
(cod. DTEMSO3)
L. 16.000

sono in vendita presso:
TECHNOCLUB
Via Rosellini, 12
20124 Milano

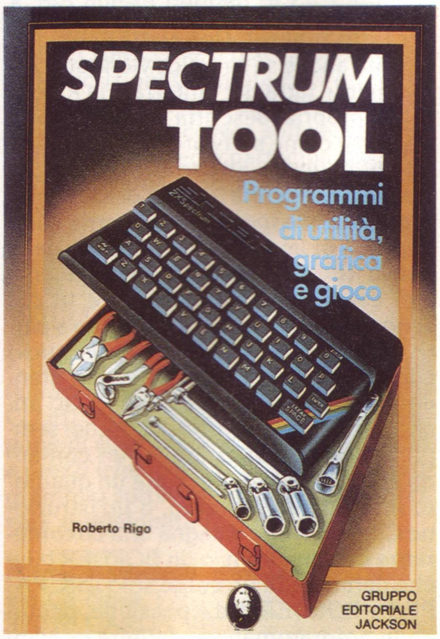
I programmi recensiti possono essere ordinati utilizzando il coupon a pag. 82

PRINT

Recensioni Libri

a cura di **Lucio Bragagnolo**

Spectrum tool
programmi di utilità, grafica e gioco di Roberto Rigo
ed. Gruppo Editoriale Jackson



Guardando la copertina del libro (per inciso, originale e simpatica) non è possibile non capirne il contenuto, anche per chi è digiuno di inglese al punto di non comprendere il titolo. Si tratta di una raccolta di programmi costituita per lo più da "strumenti" software, come dire da routine che arricchiscono lo Spectrum di funzioni e possibilità normalmente disponibili su calcolatori più potenti e costosi, ma non implementate nella ROM del nostro "piccolo" Sinclair. Vi è anche una sezione — di modeste dimensioni rispetto a quelle complessive del libro — dedicata ai giochi; ma è ben lungi dal risultare, superflua, e si dimostra utile sia per alleggerire il tono generalmente un po' rigoroso dei desti sia per evidenziare la potenza ed efficacia di alcune delle routine pubblicate nelle pagine precedenti.

Qual'è, però, il punto che realmente interessa? Il fondo di routine se ne pubblicano tante, e qualcuna in più o meno non cambiano molto la situazione... in questo libro il "punto di forza" è dato non solo dall'uso estesissimo del linguaggio macchina, ma soprattutto dall'estrema attenzione posta nel dotare ogni disassemblato di commenti finalmente comprensibili ed esaurienti per chiunque. Anche per gli inesperti, poi, c'è la possibilità di evitare di dover addentrarsi nel codice macchina per (se proprio si tratti di pigrizia all'ennesima potenza) limitarsi a copiare semplicemente i listati BASIC delle routine. Infatti, ogni programma viene dapprima presentato in BASIC, completo del linguaggio macchina inserito sotto forma di linee DATA, per poi passare all'esame del l/m attraverso il disassemblato corredato, come abbiamo detto, di ricchissime delucidazioni e spiegazioni. È stata scelta questa strada, dice l'autore nell'introduzione, allo scopo di riunire i pregi di queste due diverse modalità di programmazione pur compensandone i difetti (lento e di relativamente facile comprensione il primo difficile da capire ma molto più efficiente e veloce il secondo). E questo è il filo conduttore di tutto il libro attraverso le tre sezioni di cui è composto. La prima di esse è la piú estesa e, se vogliamo, anche la più concreta, senza la quale non avrebbe avuto senso una pubblicazione. Intitolata "utilità", comprende in senso stretto tutte le routine di cui abbiamo fino ad ora accennato, nella quasi totalità frutto di un buonissimo stile di raffinatezze sicuramente "risparmiose" in termini di memoria occupata e efficientissime in sede di controlli e sicure, ma anche piuttosto ermetiche per i non esperti (sono già arrivate in redazione alcune telefonate di aspiranti programmatori in cerca di chiarimenti). In ogni caso, il livello medio del software pubblicato si mantiene comunque alto, specialmente quanto a sinteticità ed efficienza.

Per parlare dei programmi presenti in questo "Spectrum tool" c'è solo l'imbarazzo della scelta, a partire dai "soliti" e pubblicati in mille salse (presentati però in quest'ambito con arricchimenti e maggiori opzioni) fino ad arrivare a titoli che per molti costituiranno se non una sorpresa almeno una piacevole novità. Si parte col più classico dei caricatori di linguaggio macchina, qui presentato in tre forme diverse, in grado di accettare codici esadecimali nel primo e decimali nel secondo e nel terzo; in quest'ultimo è però presente una opzione di conversione in esadecimale dei numeri, che è poi possibile far rappresentare sotto forma di linee DATA autogenerate.

Anche questa routine, come quasi tutte quelle presenti, risulta utile non solo da un punto di vista di mera utilizzazione; il libro costiuisce un'ottima occasione per aumentare la propria comprensione del funzionamento dello Spectrum in generale e, più in particolare, delle possibilità d'uso delle variabili di sistema.

Proseguendo troviamo un programma per ottenere il LIST parziale di un determinato gruppo di linee di listato, un DELETE per la cancellazione di interi blocchi di programma (certe volte veramente indispensabile) e una routine di rinumerazione, contenente la sottoroutine "Find Token", scritta in codice macchina e avente la funzione di rintracciare qualsiasi istruzione che abbia come argomento un numero di linea a 4 cifre — GO TO, GOSUB, RESTORE e così via — restituendone l'indirizzo in RAM al BASIC. Segue un'autentica delizia, reperibile a livello di home computer, crediamo, unicamente sul defunto TI99 della Texas: vale a dire l'autonumber. Si prosegue con un paio di programmi che faranno la gioia dei copiatori folli: un lettore di header, un copiatore di programmi per Spectrum 48K residente sopra RAMTOP (con versione residente sotto l'inizio dei programmi BASIC, per i possessori di Sinclair inespansi e per chi vuole riuscire a copiare anche "The Hobbit") e un paio di routine che visualizzano sullo schermo l'intensità dei segnali ricevuti dalla prese EAR dello Spectrum.

La sezione due è dedicata alla grafica e, oltre a contenere programmi di scroll pixel per pixel nelle quattro direzioni, fornisce frasi capovolte o... in diagonale (!), un ingranditore di caratteri e un editor di caratteri grafici che è sicuramente da annoverare tra i migliori pro-

grammi, sia del libro che in senso assoluto.
La terza sezione, dedicata ai giochi, ne comprende una decina, dallo standard tuttavia non eccezionale. Le uniche menzioni d'onore da dare sono per "Quasimodo", arcade in cui bisogna soccorrere la bella principessa superando gli ostacoli che i minacciano sulla nostra strada (che è poi la merlatura di un castello) e "The Wall", come dire uno degli antenati degli arcade, qui riproposto con dovizia di mezzi grafici.
Avrete già capito che il nostro giudizio globale è sostanzialmente positivo; le uniche riserve che vogliamo fare sono sui testi, farraginosi e, ci sembra di capire, non troppo curati, e sull'opportunità, data la grande quantità di linguaggio macchina presente, di fornire qualche forma di controllo sull'inserimento dei codici (tipo "checksum"), del tutto assente nel libro. Comunque, "Spectrum tool" rimane un ottimo libro, consigliabile a chi ha già qualche esperienza di programmazione.

Il libro del Microdrive Spectrum

di Ian Logan
ed. JCE



"In origine era lo Spectrum, un computer tipicamente home, ed era adatto soprattutto per giochini piú o meno intelligenti..." Probabilmente potremmo cominciare a raccontare così, in tono vagamente biblico, l'evoluzione di quel calcolatore un po' atipico che è in fondo lo Spectrum. In effetti, la mancata — o meglio tardiva — introduzione in Italia di periferiche come l'interfaccia 1 e i Microdrive, unita alla ancora attesa informatizzazione di massa (intesa come il saper usare il computer, e non come il semplice averne uno — e che differenza!), hanno portato a una sottovalutazione delle possibilità di questa macchina, tanto che sono ancora molti i padri di famiglia che pensano a un calcolatore economico negli stessi termini di un giocattolo costoso, opinione ormai non più accettata in paesi come gli USA, o la Gran Bretagna stessa. Chiaramente uno Spectrum, anche dotato di Microdrive e relativa interfaccia, farà sempre una notevole fatica a gestire una azienda: ma il punto è che, grazie a queste nuove periferiche, gli si aprono strade ben diverse (forse per certi aspetti anche più soddisfacenti) di quella giocosa, beninteso bellissima e su cui non abbiamo niente da ridire, anzi!
Ora che non è più arduo come qualche tempo fa procurarsi un Microdrive, che vengano edite anche in Italia pubblicazioni sull'argomento è cosa del tutto naturale; e finalmente veniamo a questo "Libro del Microdrive Spectrum", presentato sul retro della copertina come qualcosa che "chiunque voglia aumentare le possibilità dello Spectrum oltre quelle che già conosce, deve leggere". Una presentazione così roboante e ambiziosa come questa viene avallata dal nome dell'autore, Ian Logan, che ha partecipato in qualità di osservatore alla stesura, negli stabilimenti Sinclair, degli 8K di ROM contenuti nell'interfaccia 1. Non è certo cosa da poco (nel libro si accenna anche a notizie... non pubblicabili causa richiesta esplicita della Sinclair Research: quanti autori sono documentati fino a questo punto?); e difatti dal libro traspare una conoscenza profondissima dell'hardware — ma soprattutto del software — presente in queste ultime realizzazioni.
Anche il titolo non deve ingannare: non stiamo parlando di un semplice libro sui Microdrive, ma di una trattazione completa di quello che viene a configurarsi grazie ai Microdrive stessi ma soprattutto all'interfaccia 1, come un sistema molto più potente di quanto non fosse prima quello costituito dal solo Spectrum, con le nuove possibilità di connessione in rete locale (pensare a scuole, biblioteche, club di amatori...) e di interfacciamento tramite la porta RS232, che rende possibile il dialogo con altri calcolatori, stampanti e periferiche di ogni tipo.
A tutto questo sono dedicati i sei capitoli in cui è diviso il libro, che peraltro presenta stranamente, dopo l'introduzione... l'indice analitico.
Si parte da una descrizione generica (e da zero) di quello che viene chiamato "il sistema Spectrum esteso", per poi trattarne i vari aspetti capitolo per capitolo, più approfonditamente. Il primo elemento a essere analizzato è il BASIC esteso utilizzabile con l'interfaccia 1, di cui vengono fornite sintassi, spiegazione ed applicazione di ogni comando, a tutti i livelli. Seguono poi i Microdrive, di cui viene detto se non tutto la grandissima parte, e di cui viene descritto anche il funzionamento vero e proprio (tecnica di registrazione, densità di informazione — 500 byte per pollice di nastro —, programmi per verificare lo stato dei singoli 180/190 settori in cui è diviso ogni nastro, e altro ancora); a proposito, sapevate che una POKE alla locazione 23791 eseguita prima di dare il SAVE crea più copie dello stesso file?
Il capitolo 4 è dedicato alla rete locale, una opzione abbastanza misconosciuta ma che è in grado di offrire, se ben sfruttata, grandissime possibilità. Come al solito viene detto tutto, forse anche... troppo, visto che certi dettagli di funzionamento sono comprensibili solo da appassionati non solo di computer, ma anche di elettronica. Il linguaggio si mantiene comunque rigoroso e chiaro come all'inizio, e le maggiori difficoltà di comprensione sono controbilanciate da un intelligente uso di schemi e tabelle (peraltro presenti in gran numero per tutto il libro). Una trattazione dettagliata anche se sintetica viene successivamente fatta per l'interfaccia RS232 vera e propria, descritta sia come hardware che come software. Infine, il capitolo 6, dedicato all'uso del linguaggio macchina, che farà la gioia di tutti i programmatori esperti, messi come sono in grado di lavorare sulle routine della nuova ROM. Di esse vengono dati i codici esadecimali di accesso, e in più vengono descritte, in una sezione a parte, le modalità per ottenere dei nuovi comandi da aggiungere a quelli standard del BASIC Sinclair. Vengono anche forniti i disassemblati di alcuni semplici esempi: per esempio, un comando LINE capace di tracciare un quadrato intorno allo schermo, o un'istruzione CIRCLE n che disegna una circonferenza di diametro n, o un BORDER b,i,p che definisce in un colpo solo INK, PAPER e BORDER. Ed è possibile, per un abile programmatore, definire comandi ancora più sofisticati e potenti...
Non i pensi comunque che questo sia un libro adatto solo per "maghi del software": le messe di informazioni in esso contenuta, per ogni livello di conoscenza, è talmente alta che anche il più sprovveduto degli utenti Sinclair potrà utilizzarlo con piena soddisfazione, senza il timore di avere speso inutilmente dei soldi. Si intende che è meglio se si possiedono l'interfaccia 1 e almeno un Microdrive... ma può darsi che dopo aver terminato la lettura vi sia venuta voglia di comprare entrambi!

Spectrum Tool (cod. AJAK022) L. 13.500
Il libro del Microdrive Spectrum (cod. AJCC004) L. 14.400
sono in vendita presso: TECHNOCLUB
via Rosellini, 12 - 20124 Milano
I libri possono essere ordinati utilizzando il coupon pubblicato a pag. 82

ZX CLUB

I Club Sinclair

Prima di programmi e notizie, apriamo questa puntata di ZX Club con due annunci:
Il SINCLAIR COMPUTER CLUB ha aperto le iscrizioni (la quota è di quindicimila lire) per l'anno 1984/85. L'iscrizione dà diritto a ricevere software, il bollettino del club ed a partecipare a tutte le iniziative da esso promosse. Scrivete a:

Andrea Focardi
via G. Di Vittorio 56
50015 GRASSINA FI

A tutti gli "spectromani": ho fondato un fantastico club (il SINCLAIR SOFT-CLUB) senza fini di lucro. Esso si propone di scambiare, comprare, vendere a prezzi irrisori (L. 1000 - mille! - a programma) software per Spectrum. Scrivete o telefonate a:

Marco Maffezzoli
via Cà Nova 17
37016 GARDA VR
Tel. 045/7255097

Quanto al software, cominciamo con una sobrutine per ZX81 di Roberto Pareschi, membro del SINCLAIR CLUB BOLOGNA e residente in via Ronco 16 a Paullo (MI). Si tratta di una sobrutine di gestione di input, ma attenti! Non si tratta di un programma completo (se provate a farlo girare così com'é, non otterrete alcun risultato utile), ma va aggiunto, appunto come sobrutine, al vostro programma di inserimento dati/gestione biblioteca/agenda telefonica/fate voi. L'indirizzo del club, per chi vuole inscriversi o richiedere informazioni, è:

Sinclair Club Bologna
Associazione di Utenti di microcomputers
via Molino Vecchio 10/F
40026 IMOLA BO

Ecco le "istruzioni per l'uso":
occorre avere definito precedentemente nel vostro programma le variabili C$ (il carattere che desiderate usare come cursore) e CAMPO, che indica il numero massimo di caratteri che vengono accettati come input.
La sobrutine viene mandata in esecuzione con GOSUB 91000.
All'uscita della sobrutine abbiamo in a$ il nostro input, che definiremo come VAL A$ se l'input è numerico. Si può posizionare il cursore dove si vuole inserendo una istruzione PRINT AT N,N; prima di mandare in esecuzione la sobrutine.
Sullo schermo comparirà un cursore lampeggiante nella posizione da voi indicata
Per introdurre il dato crivetelo normalmente e poi premete NEWLINE. Per cancellare c'é il solito RUBOUT, mentre per avere uno spazio bisogna premere la freccia a destra.
Questa routine è molto utile in programmi dove viene preparata una "maschera" in cui vengono introdotti i dati. Il cursore si posizionerà di volta in volta sul campo in cui verrà introdotto il dato, passando poi automaticamente al campo successivo (per effetto di un comando NEWLINE o del raggiungimento dei limiti del campo)
Durante l'esecuzione, viene usata la variabile di sistema DF-CC, che viene decrementata dopo ogni stampa del cursore o di uno spazio. Per questo sono state usate alcune semplici istruzioni in linguaggio macchina, che dovranno essere alloggiate nella linea 1 del programma utilizzato, sotto forma di un'istruzione REM seguita da dieci caratteri. Le dieci locazioni corrispondenti sono quelle che vanno dalla 16514 alla 16523 (il codice macchina viene richiamato dai vari RAND USR 16514 presenti nella sobrutine); in esse dovranno essere inseriti tramite POKE e i seguenti valori (in decimale) 237, 75, 14, 64, 11, 237, 67, 14, 64, 201.

Subroutine di gestione input con scelta del cursore di Roberto Pareschi - Sinclair Club Bologna versione per ZX81 16K

(listato 1)

```
9000 PRINT C$;
9010 RAND USR 16514
9015 FOR K=1 TO 10
9020 LET R$=INKEY$
9025 IF R$<>"" THEN GOTO 9070
9030 NEXT K
9035 PRINT " ";
9040 RAND USR 16514
9045 FOR K=1 TO 10
9050 LET R$=INKEY$
9055 IF R$<>"" THEN GOTO 9070
9060 NEXT K
9065 GOTO 9000
9070 PRINT " ";
9075 RAND USR 16514
9080 LET K=10
```

```
9085 NEXT K
9090 LET R=CODE R$
9095 RETURN
9100 LET A$=""
9105 GOSUB 9000
9110 IF R=118 THEN RETURN
9115 IF R<>119 THEN GOTO 9165
9120 LET X=LEN (A$)
9125 IF X>0 THEN GOTO 9135
9130 GOTO 9150
9135 RAND USR 16514
9150 IF X<2 THEN GOTO 9100
9155 LET A$=A$( TO X-1)
9160 GOTO 9105
9165 IF R=115 THEN LET R$=CHR$ 0
9170 PRINT R$
9175 LET A$=A$+R$
9180 IF LEN A$>=CAMPO THEN RETUR
N
9185 GOTO 9105
```

Dal numero 5 (Marzo/Aprile 1984) del bollettino bimestrale del SINCLAIR NEW CLUB di Moncalieri pubblichiamo un programma proveniente nientepodimeno che dal... Brasile! Piuttosto ingegnoso, è cortissimo, ma funziona a meraviglia (o almeno il nostro test, effettuati sulle date della prima quindicina di agosto 1984 e sulle date di nascita dell'intero corpo redazionale, non ha rivelato errori).

Ecco l'indirizzo del club:

Sinclair New Club
c/o GianPaolo Gentili
via Turati 10
10024 MONCALIERI TO
Tel. 011/6407195

Calendario
di Carlos M. G. Weinstein - Sinclair New Club Moncalieri
versione per ZX Spectrum 16K / 48K

(listato 2)

```
 10 LET a$="144025036146"
 20 PRINT "Dammi una data, nell
    a forma       "'"~ggmmaa~ (gi
    ornomeseanno)"
 30 INPUT d$
 40 CLS
 50 LET d=VAL d$( TO 2)
 60 LET m=VAL d$(3 TO 4)
 70 LET a=VAL d$(5 TO )
 80 LET b=INT (a/4)
 90 LET x=a+b+VAL a$(m)+d
100 LET s=x-(INT (x/7)*7)
105 LET s=s+(7*(s=0))-(a-b*4=0
    AND (m=1 OR m=2))
110 PRINT AT 7,0;"Il giorno ";d.
    $;" cade di ";
120 PRINT AT 11,9;"DLMMGVS"
140 PRINT AT 12,1*s+8;"{G3}"
150 PRINT AT 20,0;"Premi un tas
    to per un'altra data"
160 PAUSE 0
170 CLS
180 GO TO 20
```

input·output

Piccoli annunci

Se volete entrare in possesso di tutte le ultime novità software per lo ZX Spectrum a prezzi irrisori (max L. 1000) scrivetemi, rispondo gratuitamente a tutti.
Vito Bianco - via Palladio, 10 - 20030 Lentate (MI) - Tel. 0362/565102

Vendo programmi per Spectrum 16K/48K, corredati di istruzioni. Scrivere allegando bollo per elenco o telefonando ore pasti a:
Maurizio Leone - via Gaio Melisso, 16 - 00175 ROMA - Tel. 06/766271

Cambio software per ZX Spectrum vasto assortimento, novità inglesi. Rispondo a tutti, inviare vostra lista, io invierò la mia.
Raffaele Maruca - via E. Toti, 27 - 21047 Saronno (VA) - Tel. 02/9606810

Vendo cassette giochi a L. 4000 (Spectrum 16K) e a L. 6000 (Spectrum 48K). Oltre 100 titoli. Inviare francobollo per listino competo. Tra i titoli (16K): Jet Pac, Psst, Tranz Am, Pac Man. Per il 48K: Ant Attak, 4D ufo attack, ecc.
Roberto Spina - C.so Italia, 304/F - 74100 Taranto - Tel. 099/331231

Cambio e vendo software per ZX Spectrum. Compro stampante ZX Printer. Telefonare dopo ottobre allo 0836/62714
Francesco Mariano - via Liguria 27 bis - 73013 Galatina (LE) - Tel. 0836/62714

Il GSINCLAIRCOMPUTERCLUB ha aperto le iscrizioni (L. 15.000) per l'anno 1984/85. L'iscrizione dà diritto a ricevere il bollettino, software ed a partecipare a tutte le iniziative promosse dal Club. Scrivere a:
Andrea Facardi - via G. Di Vittorio, 56 - 50015 Grassina (FI)

Cerco tastiera FD5 della FULLER in ottimo stato a un prezzo modico o se qualcuno sa dove trovarlo anche nuova. Inoltre vendo programmi per Spectrum 16k e 48K oppure scambio con altri programmi. Vendo anche dei programmi per Commodore CBM64, e VIC 20, oppure scambio con programmi per Spectrum. Vendo libro "alla scoperta dello ZX Spectrum" + libro dei Microdrive, e Interfaccia1, + cassetta Horizon, in Italiano a L. 30.000 valore commerciale L. 80.000 circa.
Vendo anche Videogioco Leonardo GIG o HIMEX nuovo a L. 160.000, e 5 cassette intercambiabili. Vendo antifurto per auto a tasti era digitale (senza sirena) adattabile a qualsiasi antifurto ha ben 700.000 combinazioni a L. 40.000, valore reale è di L. 180.000. Telefonare ore serali:
Valter Cacciolatto - via Doria, 21 - 10042 Nichelino (TO) - Tel. 011/6273563

Vendo per ZX Spectrum programmi tra i migliori da L. 3000 a L. 5000. Chiedete la lista o telefonate ore pasti per informazioni a:
Marco Rossi - via Cerro Maggiore - 20027 Rescaldina (MI) - Tel. 0331/57669

Vendo, ma soprattutto cambio software per Spectrum. Logicamente preferirei che qusto avenisse nella zona di Bologna ma non è necessariamente opportuno.
Pasquale Molitierno - via Silvagni, 17 - 40137 Bologna - Tel 343327

Vendo numerosi programmi a prezzi ridicoli, troppo bassi per essere pubblicati per Spectrum disponibili oltre 300 titoli 16K/48K in l/m. Scambio inoltre con software ed hardware di ogni tipo per detto computer.
Rosario Di Modica - via Castelfidardo, 37 - 97019 Vittoria (Ragusa) Tel. 0932/983512

Programmi originali per Spectrum 16/48K su cassette Sony C 60. Cambio alla pari o vendo a L. 2000 cadauno. Sconti per quantitativi. Inviate vs. lista o richiedere la mia affrancando risposta.
Cruciani Alberto - Piazza Vittorio, 138 - 00185 Roma - Tel. 06/737240

Vendo Sinclair ZX Spectrum 48K + amplificatore B.F + 21 cassette software 16/48K, il tutto a L. 500.000. Rivolgersi a:
Luca Fontana - via Garibaldi, 205/a Cornaredo (MI) - Tel. 02/9362410

Vendo, cambio numerosi programmi per ZX Spectrum 16K, giochi e utility, tra cui Jet pac, Tranz Am, Planetoids, Invaders, Poc Man, ecc. Inviare francobollo per lista. Prezzi stracciati.
Paolo Galante - viale Madonna di Rosa, 7c - 33078 S. Vito al Tagl. (ZN)

Cambio software per Spectrum. Ricerco corrsipondenti nelle varie città in cui esistano emittenti che trasmettono software.
Maurizio Monaldi - via Vittorio Montiglio, 7 - 00168 Roma

Vendo programmi per ZX Spectrum oltre 100 testi (vari) scrivere risposta a tuti. Prezzi ottimi. Avrete il listino completo.
Rosario Vicari - Via Calatafini, 977 90129 Palermo

Regalo decine di programmi per Spectrum in cambio di una stampante (ZX Printer, Alphacom 32, Seikosha GP50S). Eventualmente disponibile a L. 2500 l'uno.
Marco Sivori - Barchetta, 18/9 16162 Bolzaneto (GE) - Tel. 010/403118

Cambio Basicode 2, contro il solo rimborso delle spese invio copia manuale e del nastro di questo protocollo che permette lo scambio di programmi (Basic) tra oltre 20 diversi computer (Apple - Commodore - BBC - Sinclair - CP/M -Philips - TRS - Microprofessor, ecc.).
Maurizio Monaldi - via Vittorio Bontiglio, 7 - 00168 Roma

Vendo causa prematura fine mio ZX81, espansione memoria 16K, tastiera professionale racchiusa in mobile Teko, libri "66 programmi..." e "Imparate il basic con lo ZX81" oltre, naturalmente a moltissimi programmi.
Fabrizio Martano - via Don L. Sturzo, 7 - 58100 Grosseto - Tel. 0564/492806

Agrigento e provincia utenti computers Sinclairs cercasi per scambi di software, idee ed eventuali fondazioni Sinclub.
Giuseppe Taibi - via E. Duse, 34 - 92100 Agrigento - Tel. 71047/20768

Vendo cambio + di 250 programmi per ZX Spectrum al presso di L. 2000 a programma, risposta assicurata.
Luigi Faberi - via Campagnola, 3 - 25079 Vobarno (BS) - Tel. 0365/618388

Vendo EXP 3845, una piccola interfaccia Spectrum che ti consente di sperimentare e studiare l'hardware del computer. L. 29.000 in contrassegno.
Severi Marino - Piazza Isei, 28 - 47023 Cesena - Tel. 0547/20890

Per ZX Spectrum disponiamo di oltre 600 titoli fra cui tutte le novità inglesi. Programmi sucassette singole a L. 4000/5000 (doppia registrazione) o in sequenza su nastri di prima qualità a L. 20.000 per 6 programmi a scelta. Richiedere gratuitamente e senza impegno lista sempre aggiornata.
G. Franco Posterli - via L. Ariosto, 123 - 20099 Sesto S. Giovanni (MI) Tel. 02/2480163

Vendo molti programmi per ZX Spectrum originali inglesi a prezzi bassi,. Inoltre vendo ZX81 1K con alimentatore, libro "66 pogrammi per ZX81" ed Epron, programmata con "Space Invaders" a L. 100.000. Richiedere listino gratis.
Francesco Ficcabrino - viale Trieste, 36 - 93100 Caltanisetta - Tel. 0934/26045

Compro software per ZX Spectrum qualsiasi listato o programma. Inviatemi le vostre liste, rispondo a tutti. Cambio inoltre stupendo gioco elettronico della F1 con almeno 10 programmi compresa cassetta. Scrivete a:
Marco Gorin - C.so Italia, 57 - 13100 Vercelli

Compro, vendo, cambio software per ZX Spectrum, Apple, C64, VIC 20, TI 99/4A: Richiedre lista gratuita a:
Sergio Remondino - via De Gasperi, 3 - 10015 Ivrea (TO)

Vendo programmi per ZX Spectrum 16/48K a sole L. 2000 cadauno. Richiedere elenco scrivendo e inviando bollo per risposta e telefonare ore ufficio a:
Pietro Melissari - via Stazione, 4 - 18011 Arma di Taggia (IM) - Tel. 0184/448388

Cerco possessori di ZX Spectrum, per cambio programmi. Inviare lista, risposta assicurata.
Andrea Moggio - via N. Sauro, 15 - 13051 Biella (VC) - Tel. 015/32644

Vendo ZX Spectrum 16 + manuale italiano + "77 programmi per lo Spectrum" + "Programmazione dello Spectrum" + 85 listati + 40 programmi in cassetta, tutto a L. 300.000 + spese postali. Telefonare ore pasti a:
Luca Minuder - via Kennedy, 11 - 31015 Conegliano (TV) - Tel. 0438/34272

Cambio, vendo, compro programmi per ZX Spectrum di ogni genere. Inviare propria lista risposta assicurata oltre 200 programmi
Giulio Guidotti - Via del Popolo, 5A 56036 Palaia (PI) - Tel. 0578/622178

Cambio programmi per ZX Spectrum 16-48K. Inoltre cambio anche programmi per Commodore CBM 64. I migliori sulla piazza!
Stefano Goldoni - Via Pascal 33/2 - 4110 Modena - Tel. 059/242849

Vendo cambio software utilità, applicativo e video games per ZX Spectrum. tra i titoli hurg (generatore videogiochi) ant attack, jet set willy, printer set, personale finance, magazzino. Ma anche molti molti altri per elenco. Scrivere a:
ALberto Catania - Vie Vercelli , 21 10036 Settimo T.SE (TO) - Tel. 8009478

Costruisco interfaccia programmabile per joystick, compatibile con qualsiasi gioco per ZX Spectrum-ZX81. Possibilità di collegamento per uno o due joystick. Prezzi più che modici!!! Per informazioni telefonare a:
Fulvio Dumini - Via G. Pescetti, 79 - 50127 Firenze - Tel. 055/410823

SUPERSINC LIBRI

SPECTRUM TOOL cod. AJAK022 L. 13.500
IL LIBRO DEL MICRODRIVE SPECTRUM cod. AJACC004 L. 14.400

CEDOLA DI ORDINAZIONE - LIBRI
da compilare e spedire in busta chiusa a
J.*soft* - Via Rosellini, 12 - 20124 Milano - Tel. 6888228

Ordino i seguenti libri per un importo totale di L. .... + L. 2.000
come contributo fisso per spese di spedizione

Cod. .... Cod. .... Cod. ....

Cod. .... Cod. .... Cod. ....

Contanti allegati   Assegno allegato n° ....

Ho spedito l'importo a mezzo vaglia postale

Ho versato l'importo sul ccp. n° 19445204 intestato a J.*soft* - Milano

Pagherò in contrassegno al postino al ricevimento dei volumi (valido solo per i soci in Italia)

Nome ....

Cognome ....

Via ....

Città .... C.A.P .... Prov. ....

Se richiesta fattura:
Cod. F. e P. Iva ....

Data ....

Firma ....

Per i soci residenti all'estero — pagamento anticipato (vaglia o versamento su ns. ccp)

SUPERSINC SOFTWARE

TOWER OF EVIL cod. DTMSO2 L. 16.000
VOLCANIC PLANET cod. DTMSO3 L. 16.000

CEDOLA DI ORDINAZIONE - CASSETTE
da compilare e spedire in busta chiusa a
J.*soft* - Via Rosellini, 12 - 20124 Milano - Tel. 6888228

Ordino le seguenti cassette per un importo totale di L. .... + L. 2.000
come contributo fisso per spese di spedizione

Cod. .... Cod. .... Cod. ....

Cod. .... Cod. .... Cod. ....

Contanti allegati   Assegno allegato n° ....

Ho spedito l'importo a mezzo vaglia postale

Ho versato l'importo sul ccp. n° 19445204 intestato a J.*soft* - Milano

Pagherò in contrassegno al postino al ricevimento delle cassette

Nome ....

Cognome ....

Via ....

Città .... C.A.P .... Prov. ....

Se richiesta fattura:
Cod. F. e P. Iva ....

Data ....

Firma ....

Per i soci residenti all'estero — pagamento anticipato (vaglia o versamento su ns. ccp)

SUPERSINC INPUT/OUTPUT

La rubrica INPUT/OUTPUT è gratuita ed aperta a tutti i lettori. Chi desidera comprare, vendere o cambiare hardware o software può inviare il tagliando a J.soft - Via Rosellini, 12 - 20124 MILANO

COMPRO   VENDO   CAMBIO

ZX80   ZX81   ZX Spectrum   PERIF.   SOFTWARE

....

Nome .... Cognome ....

Via .... C.A.P ....

Città .... Tel. ....

SUPERSINC è bello, però... (ovvero suggerimenti, idee, critiche, richieste e tutto ciò che vi passa per la testa).

....

Nome .... Cognome ....

Via .... C.A.P ....

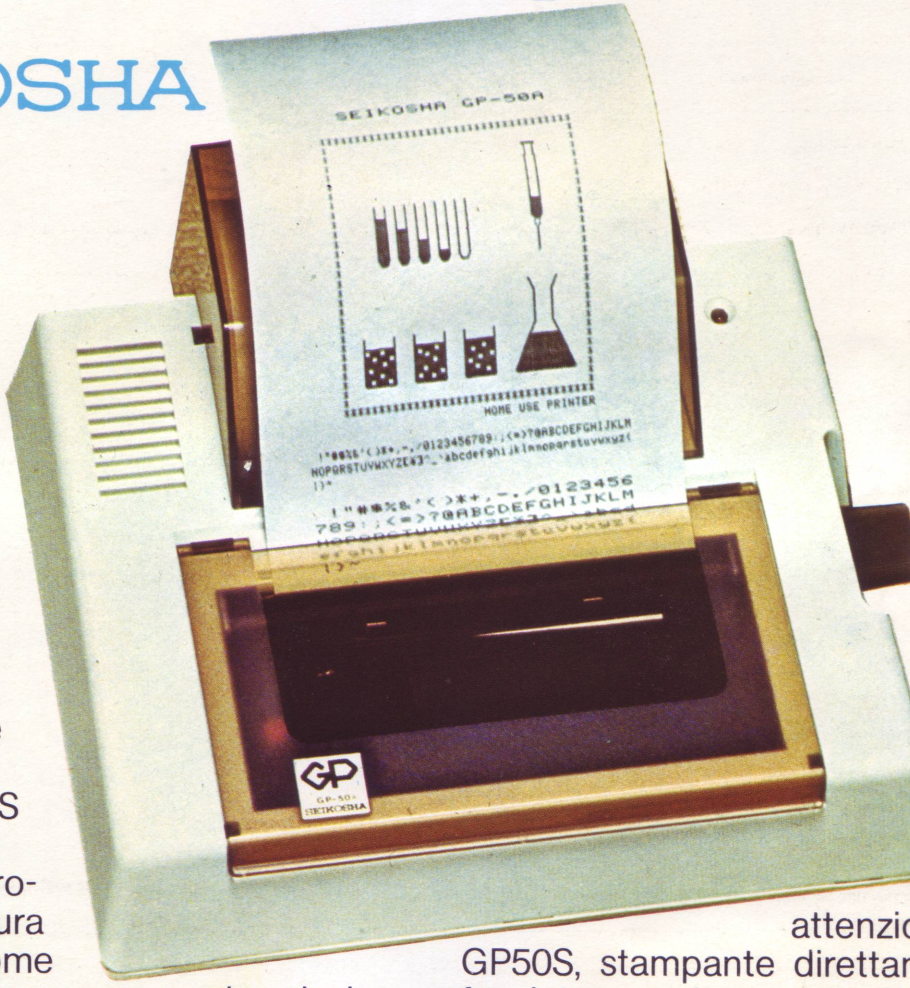
Città .... Tel. ....

GP50A E GP50S

le piccole stampanti per tutti i computer

SEIKOSHA

REBIT COMPUTER
A DIVISION OF G.B.C.



Piccole e compatte dalle prestazioni grandi e generose, le GP50A e GP50S sono realizzate con standard professionali a misura di Personal e Home computer e si impongono quale soluzione ottimale per gli usi hobbystici più disparati a costi incredibilmente sorprendenti. Particolare attenzione merita la GP50S, stampante direttamente interfacciata verso i computer Sinclair ZX81 e Spectrum.

Caratteristiche:

- Stampante ad impatto a matrice di punti da 46 colonne (32 colonne versione GP50S)
- Matrice di stampa 5x8 (7x7 versione GP50S)
- Percorso di stampa monodirezionale (da sinistra a destra)
- Capacità grafiche con indirizzamento del singolo dot
- Possibilità di ripetizione automatica di un carattere grafico
- Velocità 40 caratteri/secondo (35 caratteri/secondo versione GP50S)
- Caratterizzazione: 12 cpi e relativo espanso
- Interfacce: parallela centronics (interfaccia Sinclair versione GP50S)
- Alimentazione carta a frizione (largh. carta fino a 5")
- Stampa 1 originale e 1 copia
- Set di 96 caratteri ASCII
- Consumo 11W (standby) o 17W (stampa)
- Livello di rumore inferiore a 60 dB
- Durata di vita testa: 30 milioni di caratteri
- Peso 1,5 KG
- Dimensioni: 215 (prof.) x 250 (largh.) x 85 (alt.) mm.
- Nastro nero (standard); optionals: rosso, arancio, verde, blu, viola e marrone.

PERSONAL COMPUTER:
le prime, le migliori!

PERSONAL SOFTWARE
L'unica che presenta software
per tutti i personal:
Commodore, Apple, Sinclair, T.I., HP,
Sharp, Sega, Olivetti, ecc.



BIT: La più letta, la prima e più diffusa
TEST: Decision Mate V
SUPERBIT - 64 pagine di programmi
per i vostri personal computer.

Con tutta la competenza del
GRUPPO EDITORIALE JACKSON